SKROLLI 2019.2

SKROLLI

2019.2
SKROLLI.FI

TIETOKONEKULTTUURIN ERIKOISLEHTI

LASKENTA

GPGPU-laskenta C++/GLSL:lla
Suomen kielen sanavektorit
Rekursio kompleksi-integraalilla

SARJA JATKUU
Tee-se-itse-tietokone 1960-luvulla

Näin sinustakin tulee kasettilameri

Tietokoneet autoissa

Robottiauto. Nyt?
Neuroverkko RC-autoilee
Saabin salattu peli

MiSTer | Staattisen HTML:n paluu | Musiikkiohjelmat | eSM 2019

92 sivua ISOIN SKROLLI

Suloisen suomalainen konsepti

Pienessä maassa ja etenkin pienessä kielialueessa on omat haittansa. Varsinkin ennen internetiä jäimme usein sivuun kaikkein uusimmista virtauksista tai saimme ne vasta aikojen kuluttua jotenkin laiskasti kotoistettuina versioina. Nykyisin taas mediakentän haasteet ovat meillä moninkertaisia verrattuna maailmankieliin, joilla tavoittaa jopa satoja miljoonia ihmisiä kerralla. Muutamalle tuhannelle lukijalle kirjoittaminen vaatii vaikkapa lehden kustantajalta ihan omanlaistaan aatteen paloa.

Mainiona kääntöpuolena on kuitenkin se, että harrastuskentillä erilaiset viiteryhmät ovat tottuneet toimimaan yhdessä. Jo 1980-luvulla tietotekniikkalehdet toivat yhteen mikroraudasta kiinnostuneet rakentelijat, aloittelevat ja edistyneemmät ohjelmoijat sekä ne, jotka odottivat uusimmalta numerolta kaikkein eniten tuoreita peliuutisia ja -arvosteluja. Osittain kyseessä oli pakon sanelema ratkaisu, sillä vain yhtä näistä aiheista käsittelevä julkaisu olisi lähes väistämättä jäänyt tuolloin pienkustanteeksi. Lopputulos oli kuitenkin myönteinen, sillä näin tietystä aihealueesta kiinnostunut tuli kuin vaivihkaa lukeneeksi myös muuta. Itse en 11-vuotiaana olisi takuulla tullut selanneeksi 6502:n konekielestä kertovaa artikkelia, ellei se olisi sattunut olemaan samojen kansien välissä kuin tuoreimmat Eliten pelivinkit.

Janne Sirén kirjoittaa toisaalla lehdessä seikkailuistamme Skrollin kansainvälisen version parissa. Joukkorahoitettu 2016.1E ja ihan itse kustannettu 2017.1E eivät tehneet maailmalla suuria aaltoja, mutta avasivat silmiä puolin ja toisin: pieni ulkomaalaisten joukko näki poikkileikkauksen pienen maan omintakeisesta ja rikkaasta tietokonekulttuurin kentästä; me taas ymmärsimme, millainen outolintu Skrollin kaltainen "kaikille avoin" bitinnypläyslukemisto oikeastaan onkaan. Englanninkielisessä maailmassa kun lehdet on varsin usein tehty spesifille kohderyhmälleen.

Skrollin visiona on ollut alusta asti altistaa lukijoitamme kaikenlaiselle. Periaatteena on, että kunhan juttu on mielenkiintoisesti kirjoitettu ja sivuaa tukevasti tietotekniikkaa, se sopii meidän lehteemme. Siksi emme jää pyörimään pelkkään kuusnepaan ja Amigaan, aivan kuten emme aio siirtyä esittelemään pelkästään uusinta pilvitekniikkaakaan. Me olemme harkiten maustettu sillisalaatti kaikenlaista luettavaa tietotekniikasta ja sen ympäriltä. Harrastajilta harrastajille ja muuten vain kiinnostuneille. Ja niin on hyvä.

P.S. Viime pääkirjoituksessa kerroin, että Skrolli on ostanut hehtaarin verran ennallistettavaa suota kompensoidakseen hiilidioksidipäästöjään. Saimme paperit juuri ennen lehden painopäivää, joten nyt se on virallista: Skrollin sijoitus kohdistuu Iin Kuivaniemen Pekkassuohon.

*P.P.S. Viimeisenä päivänä ennen painoa saimme tiedon, että Mikrobitin ex-päätoimittaja **Markku Alanen** on menehtynyt. Markku oli tärkeä henkilö lehtiprojektia käynnistettäessä ja kirjoitti myös kolumnin vuoden 2013 kakkosnumeroomme. Vielä paljon suurempana tuntuu hänen vaikutuksensa siinä, mistä lähtökohdasta Skrollia on sen historian ajan toimitettu. Otamme osaa omaisten suruun ja toivotamme hyvää matkaa.*

Mikko Heinonen
vastaava päätoimittaja

4041 0209
Painotuote
HÄMEEN KIRJAPAINO OY

SKROLLI – Tietokonekulttuurin erikoislehti

Yhteydenotot	toimitus@skrolli.fi Ircnet: #skrolli skrolli.fi
Vastaava päätoimittaja	Mikko Heinonen
Päätoimittaja	Tapio Berschewsky
Toimituspäällikkö	Janne Sirén
Taitto	Manu Pärssinen Susanna Viljanmaa
Digipäällikkö, mediamyynti	Toni Kuokkanen
Kuvatoimittajat	Laura Pesola Mitol Meerna
Talous	Anssi Kolehmainen
Yhteisömestari	Mika Hyvönen
Toimitus	Jarno Niklas Alanko, Jari Jaanto, Jukka O. Kauppinen, Ronja Koistinen
Kansikuva	Laura Loukola
Sisuskuvat ▹	Rama/WmC, Laura Loukola, Janne Sirén
Tämän numeron avustajat	Lauri Anttila, Miika Auvinen, Mika Haulo, Mikael Heikkanen, Ada-Maaria Hyvärinen, Jakke, Janina Joutsen, Emma Kantanen, Tuula Keränen, Toni Kortelahti, Ninnu Koskenalho, Valhe Kouneli, Nanne Kukkura, Antti Laaksonen, Janne Laukkanen, Jarkko Lehti, Sakari Leppä, Sakari Lönn, Mikko Möykky, Tuukka Norri, Antti Nuortimo, JJ Nääs, Atte Penttinen, Mikko Pohja, Tomi Pyy, Ville Ranki, Mikko Rasa, Felipe Santana, Joseph Watson
Julkaisija	Skrolli ry
Painopaikka	Hämeen Kirjapaino oy, Tampere,
	ISSN 2323-8992 (painettu)
	ISSN 2323-900X (verkkojulkaisu)

SKROLLI 2019.2

2 **Pääkirjoitus**

4 **C++/GLSL: GPGPU-laskenta**
Pistetään grafiikkasuoritin (muihin) hommiin.

8 **Visuaalinen debuggaus**
"Print-käsky" shader-bugien metsästykseen.

10 **Rekursio kompleksi-integraalilla**
Uusi ratkaisu rekursioon.

12 **Suomen kielen sanavektorit**
Luonnollisen kielen käsittelyn salainen ase.

17 **K.O.: Tabulaattori vs. välilyönti**
Väittely koodin sisentämisestä päättyy tähän.

18 **Kesän teknolelut**
Skrolli haki Lidlistä täydennystä kesäreissuun.

20 **Kolumni: Janne Sirén**
Purkkiperspektiivi: sosiaalinen media on epäsosiaalinen.

22 **Lyhyet**
Pikkujutuissa virransäästö- ja kierrätysneuvoja.

23 **Neuroverkko pörisi**
Neurosyntetisaattori cellF miksaili aivan omiaan.

24 **Musiikkiohjelmat**
Välineet ja tekniikat tietokonemusiikin tekoon.

28 **Sirénin sisäpiiri**
Teknologiapalstalla nyt: robottiautot.

36 **Neuroverkko RC-autoilee**
Robottiautoja on monen kokoisia.

40 **Pientä PDF-muokkausta**
Kuinka vaikeaa se voi olla?

43 **Staattisen HTML:n paluu**
--- UNDER CONSTRUCTION ---

46 **Kolumni: Antti Laaksonen**
Indeksit tekevät ohjelmoinnin oppimisesta turhan vaikeaa.

47 **Ensimikro ennen mikropiirejä**
Sarja jatkuu: näin rakennettiin tietokone 1960-luvulla.

51 **Kolumni: Jarno Niklas Alanko**
Bitcoin tositilanteessa.

52 **Omenaläppärin uusiokäyttö**
Vanhalle PowerPC-Macille uusi elämä NetBSD-työasemana.

56 **Digiarkeologian alkeet**
Näin sinustakin tulee kasettilameri.

60 **Dataraita vinyylilevyllä**
Skrolli kokeilee: Miksei "LP-ROM" lyönyt läpi?

61 **Mies, joka pelasti Harvesterin**
Seikkailupelin ympärille syntyi hylkiöiden kultti.

64 **eSM 2019**
Kurkistus kotimaisen e-urheilun kulisseihin.

66 **Saabin salattu peli**
Onko Saabissa Pong?

69 **Talentec Bartop Kit**
Peliautomaattien inspiroima viihdekeskus kahvihuoneeseen.

70 **Olisinpa maailmanvaltias**
JOKin pelipalstalla kolttostellaan rajapyykeillä.

72 **MiSTer**
Kasa FPGA-klassikoita kerralla.

76 **Ei näin!**
Amiga-yhteisö osaa kaiken muita paremmin. Valitettavasti.

78 **Minikonsolit – Tiesitkö, että...**
Tärpit retropelaamisen trivial pursuitiin.

80 **Skrolli 6,4 vuotta sitten**
Skrollin uusin ja samalla vanhin palsta aloittaa – alusta.

81 **Skrolli-kalenteri**
Tiedä, mitä tietokoneen tasavallassa tapahtuu.

82 **International Edition**
Skrolli kävi ulkomailla. Miten kävi?

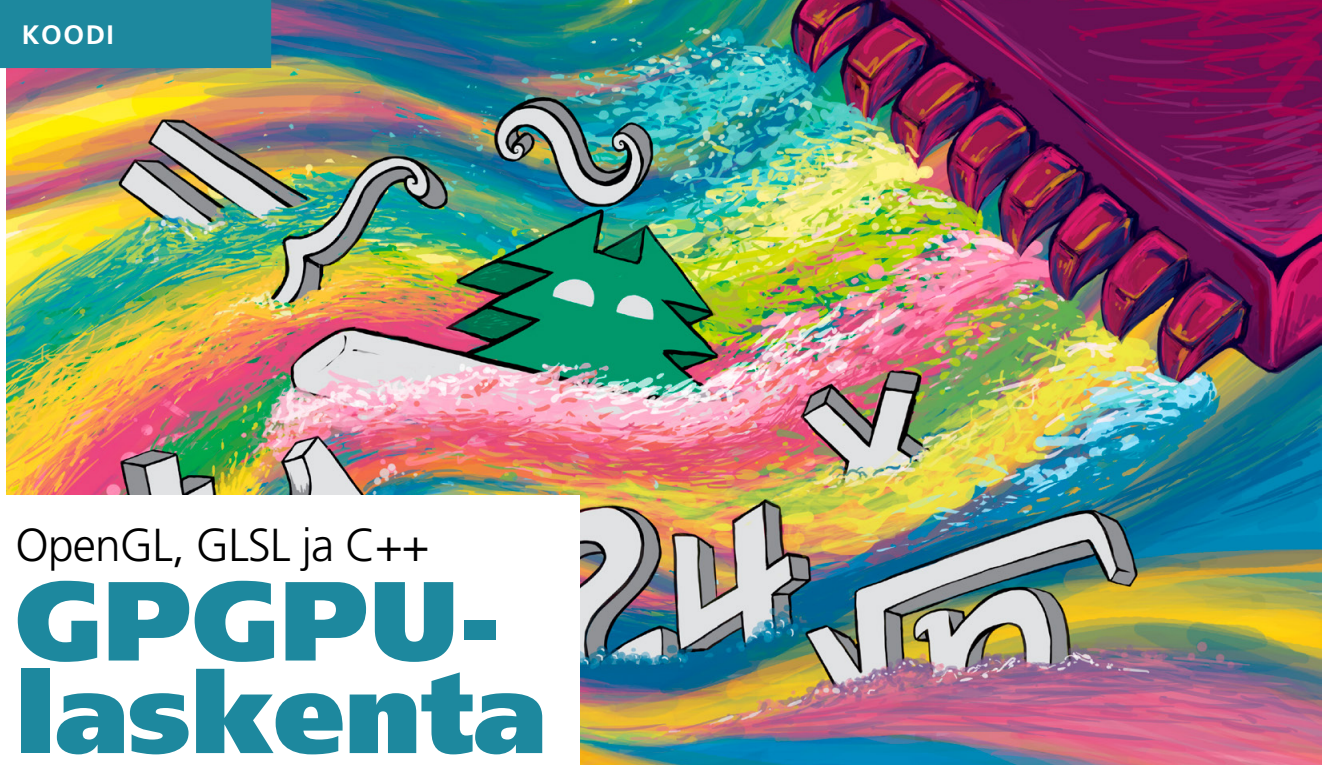
85 **Postipalsta**
Neuroverkko kirjoitti kirjeen.

86 **JOKstoriaa IX**
Grendel-hifiä ja salanimiä.

88 **Skrolleri**
Skrollin demopalstalla maailman ensimmäinen kompo.

90 **Sarjakuva**
Turrikaanien yössä seikkailupelaajan maailmantuskaa.

Skrollin kansi on tribuutti **Jean "Moebius" Giraudin** tekemälle Matra Alice -tietokoneen (1983) kansipiirrokselle. Alice on ranskalainen versio Tandy TRS-80 MC-10:stä. Piirros esiintyy sen pakkauksessa ja materiaaleissa.



OpenGL, GLSL ja C++

GPGPU-laskenta

GPGPU ei ole grafiikkasuorittimen lyhenteen kirjoitusvirhe, vaan viittaus yleiskäyttöiseen laskentaan grafiikkasuorittimella. Lähdimme seikkailulle grafiikkasuorittimen väärinkäytön maailmaan.

Teksti: Antti Nuortimo, Mikko Rasa, Jarno Niklas Alanko, Janne Sirén
Kuvat: Mitol Meerna, Antti Nuortimo

Nykyaikainen 3D-kiihdytetty näytönohjain sisältää grafiikkasuorittimen (GPU), joka on tietokonegrafiikan laskutoimituksiin suunniteltu suoritin, jossa on runsaasti rinnakkaisia ytimiä. Tällaisia tietokonegrafiikan operaatioita ovat esimerkiksi valaistuksen laskeminen ja kolmioiden kulmapisteiden koordinaattimuunnokset. Näytönohjaimien kehittyessä huomattiin kuitenkin pian, että niillä voi laskea muutakin kuin grafiikkaa. Esimerkiksi neuroverkkolaskennassa (ks. esim. Skrolli 2019.1) ja kryptovaluuttojen louhimisessa käytetään usein grafiikkasuorittimia.

Varhaiset grafiikkaan liittymättömät algoritmit GPU:lla toteutettiin grafiikkarajapinnoilla, jolloin näytönohjain tavallaan huijattiin ratkomaan ongelmia, joita sitä ei ollut suunniteltu ratkomaan. Mahdollisuudet laajenivat, kun näytönohjaimiin tuli tuki ohjelmoitaville shadereille ja liukulukutekstuureille 2000-luvun alussa. Näiden ansiosta näytönohjaimen grafiikkaliukuhihna oli nyt mahdollista valjastaa tekemään mielivaltaista laskentaa käyttämällä tekstuureja kaksisuuntaisena tiedonsiirtoväylänä CPU:n ja GPU:n välillä.

Tällaisesta laskennasta käytetään nimitystä GPGPU, eli *general-purpose computing on graphics processing units*. Myöhemmin myös näytönohjainvalmistajat tajusivat tässä piilevän potentiaalin ja lanseerasivat erilliset rajapinnat pelkästään laskentatarkoitukseen. Termi voi viitata tavallisella GPU:lla tehtävään laskentaan tai GPU-piiriä käyttävään laitteeseen, jonka pääasiallinen tarkoitus on grafiikan sijaan laskenta. Näissä laskentakorteissa ei välttämättä ole edes ulkoisia liitäntöjä näytölle.

Skrollin avustaja **Antti Nuortimo** on kehittänyt Ylikuutio-nimistä pelimoottoria (ks. Skrolli 2018.4) ja ryhtyi sen pohjalta tutustumaan GPGPU-laskentaan – sillä 2000-luvun alun tavalla. Antti kertoo nyt meille, mihin näytönohjaimen jekuttaminen johti.

Tässä artikkelissa viitataan OpenGL-grafiikkarajapintaan – aloittelevan OpenGL-ohjelmoijan kannattaa lukea ennen tätä artikkeli *Näin syntyi Assembly-demo* (Skrolli 2018.3) sekä OpenGL-kurssi Skrollin numeroista 2013.2–2014.2 (pdf-lehdet: skrolli.fi/numerot).

Tähän pystyy vanhakin

Reunaehdoikseni asetin seitsemän vuotta vanhan ThinkPad W520 -kannettavan ja sen sisäisen Intel HD Graphics 3000 -näytönohjaimen tukeman OpenGL 3.0:n. OpenGL-version 4.3 myötä tulleet niin kutsutut *compute-shaderit* eivät siis olleet käytettävissäni, joten toteuttaisin GPGPU-laskentaa vanhanaikaisesti ohjelmoitavilla sharedeilla käyttämällä shaderien ohjelmointikieltä OpenGL Shading Language (GLSL) ja datan tallentamiseen tekstuureita. Pyrin myös välttämään kaikkia OpenGL-laajennuksia.

Löysin vuodelta 2005 peräisin olevan GPGPU-tutoriaalin (linkit verkkojatkoilla), joka on jo hieman vanhentunut. Vaatimuksena mainitut Nvidia GeForce FX ja ATI Radeon 9500 tukevat korkeintaan OpenGL 2.1:tä. Pelimoottoriprojektini ruutukaappaustoiminnon peruina tiesin, että OpenGL 3.0:lla on mahdollista piirtää kuva näytön ulkopuoliseen ruutupuskuriin (*offscreen framebuffer*) ja kopioida puskurin sisältö keskusmuistiin käyttämättä yhtäkään OpenGL-laajennusta. Päättelin tästä, että OpenGL 3.0:lla on mahdollista suorittaa GPGPU-laskentaa ilman OpenGL-laajennuksia.

Tämän tekniikan etuna on, että koodi toimisi todella laajalla joukolla koneita – myös vähän vanhemmissa kannettavissa. Käsittelen asiaa tässä artikkelissa pääasiassa irrallaan muusta ohjelmakoodista, mutta taustatietona kerrottakoon, että päätin toteuttaa GPGPU-shaderit osaksi pelimoottorini piirtoliukuhihnaa (*rendering pipeline*) olemassa olevaa koodia hyödyntääkseni. Osa pelimoottorini olioista tulisi olemaan GPGPU-shadereita, joilla on lapsiolioina laskentatehtäviä.

OpenGL on joka tapauksessa tilakone, joten samaan aikaan näytönohjaimella ei ole mahdollista suorittaa kuin

yhtä shaderia kerrallaan, toki näytönohjaimen lukuisissa ytimissä (*core)* samanaikaisesti.

Valmistellaan tekstuurit

Tavoitteena on, että laskentaolioluokka sallisi mahdollisimman joustavasti erilaisten laskentatehtävien lisäämisen. Koska tavoitteenamme on huijata GPU:ta laskemaan laskumme, naamioimme laskutoimituksen tekstuurien käsittelyksi. Kullakin laskentatehtävällä on kaksi OpenGL-tekstuuria: lähdetekstuuri ja kohdetekstuuri.

Syöte tallennetaan lähdetekstuuriin. Lähdetekstuuri sisältää käsiteltävän datan – mitä ikinä olemmekaan laskemassa GPU:lla – esimerkiksi etäisyysmatriisin Floydin-Warshallin algoritmia varten.

Datan tallentamiseen ja käsittelyyn käytetään näytönohjaimen puolella ruutupuskuriobjektiin liitettyjä tekstuureita. Tekstuurit ovat siis taulukoita ja tekselit (tekstuurin osat) ovat tämän taulukon alkioita. Tekstuurit ohjelmoin ladattavaksi 24-bittisistä BMP-tiedostoista, 32-bittisistä DDS-tiedostoista sekä CSV-tiedostoista.

Olennaista on, minkälaista dataa taulukkona toimivaan tekstuuriin on mahdollista tallentaa, eli tekstuurin formaatti ja tyyppi. Formaatti tarkoittaa pääasiassa sitä, montako lukuarvoa kuhunkin alkioon tallennetaan, ja mitä väriarvoja ne määrittävät. Esimerkiksi muoto GL_RGB tarkoittaa kolmea lukuarvoa: punainen, vihreä ja sininen. Tyyppi taas tarkoittaa kunkin lukuarvon tietotyyppiä sovelluksen puolella.[1]

OpenGL:n rajoituksia

Koska OpenGL on grafiikkarajapinta, sen laskentakäyttöön liittyy rajoituksia. Ensinnäkin laskentaan käytettävä data on jotenkin saatava välitettyä shadereille. OpenGL 3.0:ssa ainoa vapaasti indeksoitavissa oleva tietorakenne on tekstuurit.

Tekstuurit ovat yksi-, kaksi- tai kolmiulotteisia taulukoita. Jokaisessa taulukon elementissä eli tekselissä voi olla yhdestä neljään lukuarvoa. Grafiikkakäytössä nämä arvot ovat punainen (R), vihreä (G) ja sininen (B) värikanava sekä läpinäkyvyys (A). Laskentakäytössä perinteisellä tulkinnalla ei kuitenkaan ole merkitystä ja kanavia voi käsitellä mielivaltaisina arvoina. Tekstuureihin voi tallentaa myös syvyys- tai sapluuna-arvoja (*depth*, *stencil*), mutta näistä ei juuri ole iloa laskennassa.

Luotaessa tekstuuria on määriteltävä, mitä formaattia datan tallentamiseen käytetään. Perusformaatit ovat järjestyksessä yhdestä neljään komponenttiin GL_RED, GL_RG, GL_RGB ja GL_RGBA. Näitä formaatteja sellaisenaan käytettäessä OpenGL-implementaatio voi itse valita tallennustarkkuuden.[2]

Lisäksi on määriteltävä ohjelman muistissa olevan lähdedatan tallennusmuoto. Siihen käytetään historiallisista syistä perusformaatin ja datatyypin yhdistelmää.[3] Tässä on oltava tarkkana, koska jos formaatit eivät täsmää, tuloksena voi olla korruptoitunutta dataa tai kutsu voi jopa palauttaa virheen.

Laskentaa suoritettaessa shaderi voi lukea arvoja mielivaltaisista tekseleistä, mutta kirjoitus onnistuu ainoastaan kulloinkin käsiteltävään ruutupuskuriobjektiin liitetyn tekstuurin tekseliin. Algoritmin on siis oltava luonteeltaan sellainen, että jokainen tulosarvo on laskettavissa erillään muista. Jos sama tekstuuri on samaan aikaan sekä syötteenä että piirron kohteena, lukuoperaatiot tuottavat määrittelemättömiä tuloksia.

OpenGL 3.0 tukee shadereissa ainoastaan yksinkertaisen tarkkuuden (32 bittiä, float) ja puolikkaan tarkkuuden (16 bittiä, half-float) liukulukuja.[4] Kaksinkertaisen tarkkuuden (64 bittiä, double) liukuluvut ovat käytettävissä vasta OpenGL 4.0:ssa, ja silloinkin laskenta niillä on huomattavasti hitaampaa, ellei käytössä ole erityisesti laskentaan tarkoitettu näytönohjain.

Jotta laskennasta olisi mitään iloa, tulokset on vielä saatava takaisin CPU:lle. Tähän käytetään nimensä mukaisesti OpenGL:n `glGetTexImage`-funktiota. Toinen vaihtoehto on `glReadPixels`, joka lukee dataa ruutupuskurista ja pystyy lukemaan myös osittaisen alueen. Aiemmin mainitut rajoitukset formaattien yhteensopivuudesta pätevät myös näihin funktioihin.

Datan siirrossa kumpaan tahansa suuntaan on syytä huomioida myös rivien pakkaus. Oletusarvoisesti rivin pituus tavuina tasataan neljällä jaolliseksi lisäämällä loppuun tarvittava määrä tyhjiä tavuja. Jos tätä ei ole otettu huomioon datapuskuria varatessa, seurauksena on ylivuoto ja mahdollisesti ohjelman kaatuminen.[5]

Lasketaan tekstuureita

Piirtokohteena on kohdetekstuuri, tässä tapauksessa kaksiulotteinen olio, kahdesta suorakulmaisesta kolmiosta koostuva neliö, jossa käytän OpenGL:n GL_TRIANGLE_STRIP-primitiiviä.

---

[1] *Tyyppi voi toimia myös vihjeenä OpenGL:lle halutusta sisäisestä formaatista, jos parametrina annettu sisäinen formaatti on kooton (unsized), mutta toteutuksen ei ole pakko noudattaa vihjettä. Jos formaatti on GL_RGB, sisäinen formaatti myös GL_RGB ja tyyppi GL_FLOAT, tuloksena voi silti olla tekstuuri, jonka sisäinen formaatti on GL_RGB8. Esimerkiksi jos tyypiksi asetetaan GL_UNSIGNED_BYTE, kukin lukuarvo on 8-bittinen etumerkitön kokonaisluku.*

[2] *Kustakin perusformaatista on myös muunnelmia, joissa tarkkuus on erikseen mainittu. Esimerkiksi GL_R16F on yksikanavainen formaatti 16-bittisillä liukulukuarvoilla ja GL_RGB32UI sisältää kolme kanavaa 32-bittisiä etumerkittömiä kokonaislukuja.*

[3] *Esimerkiksi GL_RG, GL_UNSIGNED_SHORT tarkoittaa samaa kuin sisäisenä formaattina käytetty GL_RG16. Kuitenkin kokonaislukuformaateille on tässä yhteydessä olemassa erilliset _INTEGER-päätteellä varustetut perusformaatit erotukseksi vanhempaa perua olevista normalisoiduista reaaliluvuista.*

[4] *OpenGL-standardi edellyttää, että liukuluvut on toteutettu normaalien lukuarvojen osalta IEEE-754-standardin mukaisesti. Erikoistilanteet eli laskutoimitukset Inf- ja NaN-arvoilla (äärettömyys ja epäluku) on kuitenkin jätetty määrittelemättömiksi, joten niissä voi olla eroja implementaatioiden välillä.*

[5] *Tasausta voi muuttaa* `glPixelStorei`*-funktion arvoilla GL_UNPACK_ALIGNMENT (datan siirto CPU:lta GPU:lle) ja GL_PACK_ALIGNMENT (vastakkainen suunta). Tasausarvon pienentäminen saattaa hidastaa toimintaa.*

Vinkki: Jos jokin muuttuja ei vaikuta shaderin lopputulokseen, OpenGL-implementaatio voi optimoida sen pois. Tällöin `glGetAttribLocation` *tai* `glGetUniformLocation` *palauttaa -1, eli muuttujaa ei ole olemassa. On siis syytä tarkistaa näiden funktioiden palauttamat arvot ennen kuin niitä käyttää muissa funktiokutsuissa.*

	Haapamäki	Iisalmi	Joensuu	Jyväskylä	Kemi	Kemijärvi	Kerava	Kokemäki	Kolari	Kontiomäki	Kotkan satama	Kouvola	Kuopio	Lahti	...
Haapamäki	0	9999	9999	78	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	...
Iisalmi	9999	0	9999	9999	9999	9999	9999	9999	9999	109	9999	9999	85	9999	...
Joensuu	9999	9999	0	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	...
Jyväskylä	78	9999	9999	0	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	...
Kemi	9999	9999	9999	9999	0	9999	9999	9999	209	9999	9999	9999	9999	9999	...
Kemijärvi	9999	9999	9999	9999	9999	0	9999	9999	9999	9999	9999	9999	9999	9999	...
Kerava	9999	9999	9999	9999	9999	9999	0	9999	9999	9999	9999	9999	9999	75	...
Kokemäki	9999	9999	9999	9999	9999	9999	9999	0	9999	9999	9999	9999	9999	9999	...
Kolari	9999	9999	9999	9999	209	9999	9999	9999	0	9999	9999	9999	9999	9999	...
Kontiomäki	9999	109	9999	9999	9999	9999	9999	9999	9999	0	9999	9999	9999	9999	...
Kotkan satama	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	0	52	9999	9999	...
Kouvola	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	52	0	9999	62	...
Kuopio	9999	85	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999	0	9999	...
Lahti	9999	9999	9999	9999	9999	9999	75	9999	9999	9999	9999	62	9999	0	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Yläkulma 32 x 32 -etäisyysmatriisista, joka määrittää rautatieyhteyksien verkon listaukselle 2. Kaaret naapurisolmuihin on punaisella, etäisyydet kilometreinä. 9999 on puuttuva yhteys.

Koska ei ole tarvetta projisoida tätä neliötä 3D-avaruuteen – mehän vain leikimme tekevämme grafiikkaa – X- ja Y-koordinaatit riittävät kulmapisteiden koordinaateiksi.

Pelimoottorissani kaikki saman shaderolion lapsina olevat laskentatehtävät käyttävät laskentaan samaa GLSL-verteksishaderia ja samaa GLSL-fragmenttishaderia, mutta kullakin laskentatehtävällä on oma datansa, oma OpenGL-ruutupuskuri-olionsa ja OpenGL-tekstuurinsa dataansa varten.

Koordinaattien on oltava NDC-avaruudessa (*normalized devices coordinates*), eli vasen alanurkka on (-1, -1) ja oikea ylänurkka on (1, 1). Mitään projektioita ei tässä tarvita, koska ainoa fragmenttishaderille välitettävä data on tekstuurikoordinaatit, joita kutsutaan yleisesti U:ksi ja V:ksi erotuksena avaruuskoordinaatteina käytetyistä X:stä ja Y:stä. Lisäksi tarvitaan lähdetekstuuri ja sitä varten nämä tekstuurikoordinaatit U ja V. Indeksipuskuria ei käytetä, sillä tässä on kulmapisteitä (verteksejä) vain 4 kappaletta. Käytännössä piirrämme framebuffer-objektiin liitettyyn kohdetekstuuriin.

Piirtämistä varten asetetaan lähdetekstuuri näytteistimen (*texture sampler*) käyttämäksi tekstuuriksi ja kohdetekstuuri asetetaan rendauksen kohteeksi. Kun lähdetekstuuri ja kohdetekstuuri ovat saman kokoisia, näytteistimellä on mahdollista kuvata kukin lähdetekstuurin tekseli täsmälleen yhteen kohdetekstuurin tekseliin, eli bijektio lähdetekstuurista kohdetekstuuriin on mahdollinen.

Toki kukin fragmenttishaderin ilmentymä voi shader-koodissa lukea myös esimerkiksi viereisten tekselien arvoja. Kukin fragmenttishaderin ilmentymä voi kuitenkin asettaa vain oman tekselinsä väriarvot kirjoittamalla ne asettamaansa tulosmuuttujaan (`out` tai `gl_FragColor`-muuttujaan, jota tosin uudemmissa OpenGL-versioissa pidetään jo vanhentuneena). Mikäli tulosmuuttujaan tallentaa värikanavia, joita tekstuuri ei formaattinsa perusteella sisällä, OpenGL-implementaatio jättää kyseiset ylimääräiset värikanavat huomiotta.

Niinpä jos tekstuurin sisäinen formaatti on esimerkiksi GL_R8, tulosmuuttujan värikanavista vihreää, sinistä ja läpinäkyvyyttä ei tallenneta tulostekstuuriin.

Ensimmäinen yritys

Ensimmäisenä laskentashaderina toteutin identiteettioperaattorin, joka nimensä mukaisesti ainoastaan kopioi lähdetekstuurin sisällön kohdetekstuuriin. Selvyyden vuoksi jätän tästä eteenpäin pelimoottoriani koskevat yksityiskohdat pois ja keskityn itse asiaan.

Identiteettioperaattori GLSL:llä on hyvin yksinkertainen, mutta sillä voi testata, että kaikki toimii. Toiminta edellyttää, että lähde- ja kohdetekstuuri ovat keskenään yhtä leveitä ja yhtä korkeita, ja että niissä on sama sisäinen formaatti ja sama tyyppi:

```
varying vec2 UV;

uniform sampler2D texture_sampler;

out vec4 color;

void main() {
    color = texture(texture_sampler, UV);
}
```

Seuraavaksi lisäsin Sobel-operaattorin, jota käytetään kuvankäsittelyssä äärireviivojen tunnistamiseen. Sobel-operaattorissa kullekin tekselille lasketaan vaaka- ja pystysuuntaiset liukumat. Liukumien (*gradient*) laskenta perustuu konvoluutioon, eli kumpikin liukuma on ympäröivien kahdeksan tekselin painotettu summa, eri tavoin painotettuna. Tämän jälkeen liukuman magnitudi saadaan selville laskemalla gradienttien neliöiden summan neliöjuuri – katso listaus 1.

Iteroidaan tekstuureita

Toisinaan laskennassa joudutaan iteroimaan. Lopetusehto voidaan antaa takaisinkutsufunktiona (*callback function*). Lopetusehdon lisäksi määritellään iterointikertojen maksimimäärä. Mikäli lopetusehtoa ei ole annettu, määräksi annetaan iterointikertojen tarkka lukumäärä. Usein onkin luultavasti käytännöllisintä käyttää kiinteää iterointikertojen lukumäärää, koska ainoa keino välittää informaatiota ulos näytönohjaimelta shader-koodista prosessorille on tallentaa tulos kohdetekstuuriin.

Kun laskentaa iteroidaan, lähde- ja kohdetekstuuri vaihdetaan päittäin, jotta nykyisen iteraation kohdetekstuuri tulee asetetuksi seuraavaa iteraatiota varten lähdetekstuuriksi ja nykyisen iteraation lähdetekstuuri seuraavan iteraation kohdetekstuuriksi. Kun iterointi lopetetaan, tulos on valmiina kohdetekstuurissa.

Tämä yksinkertainen esimerkki iteroinnista siirtää tekstuuria:

```
varying vec2 UV;

uniform sampler2D texture_sampler;

uniform int screen_width;

out vec4 color;

void main() {
    vec2 UV_east = UV + vec2(1.0f / screen_width, 0.0f);
    color = texture(texture_sampler, UV_east);
}
```

Itse käytän tässä pelimoottoriini toteuttamaani takaisinkutsutoiminnallisuuden sisältävää luokkaa `CallbackEngine`. Iterointiprosessia on myös mahdollista muokata määrittelemällä kunkin iteraation alussa suo-

```
#version 130
varying vec2 UV;
uniform sampler2D texture_sampler;

uniform int screen_width;
uniform int screen_height;

out vec4 color;

void main()
{
    vec2 UV_northwest = UV + vec2(-1.0f / screen_width, 1.0f / screen_height);
    vec3 northwest = texture(texture_sampler, UV_northwest).rgb;
    vec2 UV_north = UV + vec2(0.0f, 1.0f / screen_height);
    vec3 north = texture(texture_sampler, UV_north).rgb;
    vec2 UV_northeast = UV + vec2(1.0f / screen_width, 1.0f / screen_height);
    vec3 northeast = texture(texture_sampler, UV_northeast).rgb;

    vec2 UV_west = UV + vec2(-1.0f / screen_width, 0.0f);
    vec3 west = texture(texture_sampler, UV_west).rgb;
    vec2 UV_east = UV + vec2(1.0f / screen_width, 0.0f);
    vec3 east = texture(texture_sampler, UV_east).rgb;

    vec2 UV_southwest = UV + vec2(-1.0f / screen_width, -1.0f / screen_height);
    vec3 southwest = texture(texture_sampler, UV_southwest).rgb;
    vec2 UV_south = UV + vec2(0.0f, -1.0f / screen_height);
    vec3 south = texture(texture_sampler, UV_south).rgb;
    vec2 UV_southeast = UV + vec2(1.0f / screen_width, -1.0f / screen_height);
    vec3 southeast = texture(texture_sampler, UV_southeast).rgb;

    float x_red_value = -1.0f * northwest.r - 2.0f * west.r - 1.0f * southwest.r
+ 1.0f * northeast.r + 2.0f * east.r + 1.0f * southeast.r;
    float x_green_value = -1.0f * northwest.g - 2.0f * west.g - 1.0f *
southwest.g + 1.0f * northeast.g + 2.0f * east.g + 1.0f * southeast.g;
    float x_blue_value = -1.0f * northwest.b - 2.0f * west.b - 1.0f *
southwest.b + 1.0f * northeast.b + 2.0f * east.b + 1.0f * southeast.b;
    float x_gray_value = (x_red_value + x_green_value + x_blue_value) / 3.0f;

    float y_red_value = -1.0f * northwest.r - 2.0f * north.r - 1.0f *
northwest.r + 1.0f * southeast.r + 2.0f * south.r + 1.0f * southeast.r;
    float y_green_value = -1.0f * northwest.g - 2.0f * north.g - 1.0f *
northwest.g + 1.0f * southeast.g + 2.0f * south.g + 1.0f * southeast.g;
    float y_blue_value = -1.0f * northwest.b - 2.0f * north.b - 1.0f *
northwest.b + 1.0f * southeast.b + 2.0f * south.b + 1.0f * southeast.b;
    float y_gray_value = (y_red_value + y_green_value + y_blue_value) / 3.0f;

    float gradient_magnitude = sqrt(x_gray_value * x_gray_value + y_gray_value
* y_gray_value);

    color = vec4(gradient_magnitude, gradient_magnitude, gradient_magnitude,
0.0f);
}
```

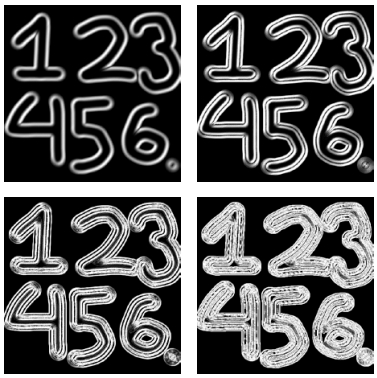
Listaus 1.

```
#version 130
varying vec2 UV;
uniform sampler2D texture_sampler;

uniform int screen_width;
uniform int screen_height;

uniform int iteration_i;

out float distance;

void main() {
    float i = UV.x;
    float j = UV.y;
    float k = float(iteration_i) / float(screen_width);

    vec2 UV_i_j = vec2(i, j);
    vec2 UV_i_k = vec2(i, k);
    vec2 UV_k_j = vec2(k, j);

    float dist_i_j = texture(texture_sampler, UV_i_j).r;
    float dist_i_k = texture(texture_sampler, UV_i_k).r;
    float dist_k_j = texture(texture_sampler, UV_k_j).r;

    if (dist_i_j > dist_i_k + dist_k_j) {
        dist_i_j = dist_i_k + dist_k_j;
    }

    distance = dist_i_j;
}
```

Listaus 2.

ritettava `PreIterateCallback` sekä kunkin iteraation lopussa suoritettava `PostIterateCallback`. Myös nämä ovat takaisinkutsufunktioita.

Esimerkiksi Sobel-operaattori (listaus 1) näyttäisi iteroituna seuraavanlaiselta:

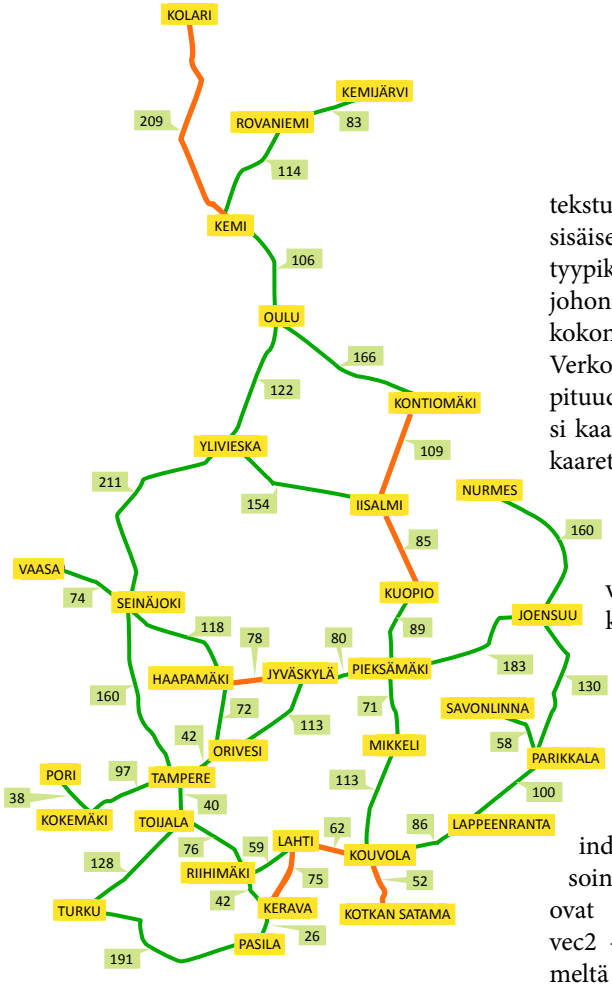


Floyd-Warshall-algoritmi

Tarkkaavainen lukija toteaa tässä vaiheessa, että eihän tämä ole GPGPU:ta laisinkaan, vaan visuaalisten efektien ohjelmointia – mihin OpenGL/GLSL on ensisijaisesti suunniteltukin. Tämä on totta. Varmistettuamme ja opastettuamme GPGPU:n kannalta olennaiset OpenGL:n ohjelmointitekniikat visuaalisten efektien avulla siirrymmekin nyt yleispätevään laskentaan.

Esimerkiksi valitsin Floydin-Warshallin algoritmin, joka on tehokas menetelmä lyhimpien polkujen löytämiseksi jonkin verkon (vaikkapa kartan) kaikkien solmujen (vaikkapa kaupunkien) välille. Algoritmin aikavaativuus on $O(|V|^3)$, missä $|V|$ on verkon solmujen lukumäärä. Floydin-Warshallin algoritmi on varsin yksinkertainen ja koostuu kolmesta sisäkkäisestä silmukasta – pseudokoodina:

```
olkoon etäisyys |V| × |V| etäisyysmatriisi, jonka kaikkien alkioiden arvot on
alustettu äärettömiksi
// Alustetaan etäisyysmatriisi.
toista jokaiselle kaarelle (u, v)
   etäisyys[u, v] = paino(u, v)
   // kaaren (u, v) paino eli kaaren
      pituus
// Varsinainen Floydin-Warshallin algo-
ritmi alkaa tästä.
toista k luvusta 1 lukuun |V|
   toista i luvusta 1 lukuun |V|
      toista j luvusta 1 lukuun |V|
         jos (etäisyys[i][j] >
         etäisyys[i][k] +
         etäisyys[k][j])
            etäisyys[i][j] =
            etäisyys[i][k] +
            etäisyys[k][j]
```



Rautatieasemista ja junayhteyksistä muodostettu verkko listaukselle 2. Punaisella ovat kaaret, joiden painot eli etäisyydet löytyvät edellisen aukeaman etäisyysmatriisista.

GPGPU-laskennan suuri juju on GPU:n rinnakkaislaskentakyvyissä. Floydin-Warshallin algoritmi soveltuu rinnakkaistettavaksi siten, että sen kahden sisemmän silmukan kaikki iteraatiot tulevat suoritetuksi rinnakkaisesti mielivaltaisessa järjestyksessä näytönohjaimen ytimissä. Näin ollen kullakin iteraatiolla indeksin k arvo on sama kaikille rinnakkaisesti suoritettaville fragmenttishaderin instansseille, ja OpenGL välittää kutakin tekseliä vastaavat interpoloidut UV-koordinaatit yhdelle fragmenttishaderi-instanssille.

Etäisyydet tulee kuitenkin lukea eri tekstuurista kuin mihin ne tallennetaan. Kun iteraation lopussa lähde- ja kohdetekstuuri vaihdetaan päittäin, edellisellä iteraatiolla kohdetekstuuriin laskettuja uusia minimietäisyyksiä käytetään seuraavan iteraation syötteenä, kun kohdetekstuurista tulee uusi lähdetekstuuri ja päinvastoin.

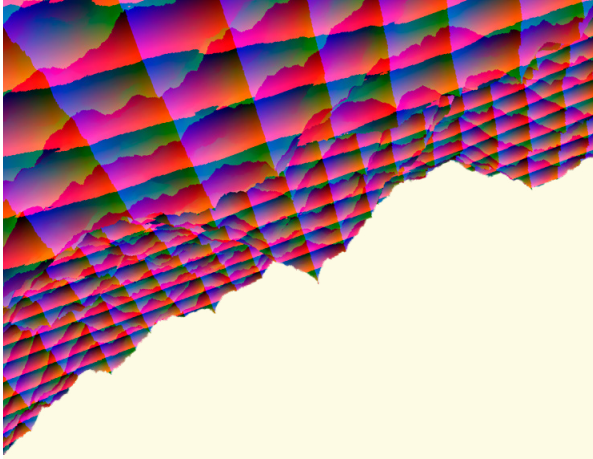
Verkko-aineistoksi otin Suomen joidenkin rautatieasemien välisiä rautatie-etäisyyksiä kilometreinä Suomen matkustajajunaliikenteen käyttämältä rataverkolta (katso kuvat) ja tallensin datan etäisyysmatriisina CSV-tiedostoon. Esimerkkitapauksessani valitsin tekstuurien formaatiksi GL_RED, sisäiseksi formaatiksi GL_R16, ja tyypiksi GL_UNSIGNED_SHORT, johon voi tallentaa etumerkittömiä kokonaislukuja väliltä 0–65 535. Verkon jokaisen puuttuvan kaaren pituudeksi syötin 9 999. Puuttuviksi kaariksi merkitsin siis kaikki ne kaaret kahden solmun välillä, joiden välillä ei ole tällä hetkellä muuta matkustajajunayhteyttä kuin jonkin tämän verkon kolmannen solmun kautta.

Laskenta on listauksessa 2. Käytän Floydin-Warshallin k-indeksinä uniform-muuttujaan iteration_i tallennettua iteraatioindeksiä. Koska Floyd-Warshallissa k-indeksin kanssa taulukon indeksointiin käytetyt muuttujat i ja j ovat fragmenttishaderin varying vec2 -tyyppisessä muuttujassa nimeltä UV välillä [0, 1], muunnan int-tyyppisen iteraatioindeksin float-liukuluvuksi jakamalla sen tekstuurin leveydellä. Tällä tavoin saan säädettyä UV-koordinaatit välille [0, 1], kun iteraatioita on enintään tekstuurin leveyden eli verkon solmujen lukumäärän verran. Tekstuurista on myös mahdollista hakea tietty tekseli kokonaislukukoordinaateilla texelFetch-funktiolla. Funktion kanssa täytyisi vastaavasti skaalata UV-muuttujasta luetut i ja j kokonaisluvuiksi.

Onnea matkaan!

OpenGL:ssä on huomattavasti erilaisia rajoitteita, mutta vanhemmallakin versiolla voi laskea esimerkiksi verkkoalgoritmeja. GLSL:llä koodatessa ei kuitenkaan kannata tehdä ilman tietoa kovin vahvoja oletuksia siitä, mikä on mahdollista. Ennen kuin ryhtyy tekemään mitään monimutkaisempaa, kannattaa harjoitella yksinkertaisilla shader-ohjelmilla.

Artikkelin listaukset, linkit ja lisätiedot Skrollin verkkojatkoilla:
skrolli.fi/numerot



Visuaalinen DEBUGGAUS

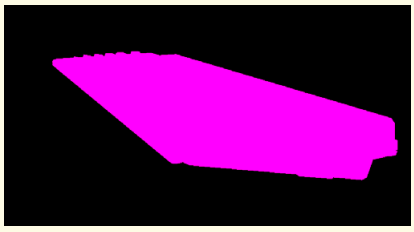
Teksti ja kuvat: Mikko Rasa

Grafiikkaohjelmoinnissa käytetyt shaderit ovat ohjelmakoodia, joten niihin voi pujahtaa ohjelmointivirheitä. Koodaajan perustyökalupakista löytyvät keinot eivät kuitenkaan pure niihin, koska debuggerilla ei pääse kiinni shaderin suoritukseen eikä GLSL:ssä liioin ole print-käskyä. Kuinka siis selvittää ja korjata virheet?

Helpoin virheluokka ovat syntaksi- ja semantiikkavirheet, joista grafiikka-ajureiden shader-kääntäjä antaa virheilmoituksen. Ne eivät ole kovin mielenkiintoisia, joten emme käsittele niitä tässä sen enempää.

Joskus voi käydä niin, ettei piirrettävä esine lainkaan ilmesty ruudulle. Onko koordinaattimuunnokset laskettu oikein? Piirtyvätkö kaikki pikselit samalla mustalla värillä kuin tausta? Ensimmäinen askel virheen selvityksessä on usein korvata fragmenttishaderi yksinkertaisella versiolla, joka piirtää vain yhtä väriä.

```
out vec4 frag_color;
void main()
{
frag_color = vec4(1.0, 0.0,
1.0, 1.0);
}
```

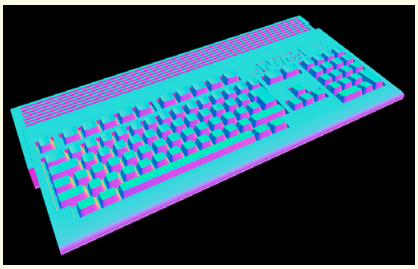


Lopputulos voi olla jotain yllä olevan kaltaista. Jos vieläkään ei näy mitään, tarkista projektio- ja malli-/näkymämatriisit sekä verteksishaderissa suoritettava laskenta.

Kun esine on saatu näkymään ruudulla, voidaan toden teolla alkaa selvittämään shadereissa tapahtuvaa laskentaa. Tekstimuotoisen tulostuskomennon puuttuessa on hyödynnettävä shaderin ruudulle piirtämää väriä. Kolme värikanavaa tarjoaa periaatteessa mahdollisuuden visualisoida kolmea eri asiaa, mutta kannattaa huomioida ihmisaivojen hahmotuskyky. Pehmeässä väriliu'ussa on vaikea havaita pieniä eroja, mutta voimakkaat kontrastit erottuvat selvästi. Toisaalta jos jokainen pikseli on eri värinen, seurauksena on kohinaa, josta ei saa tolkkua.

Eräs hyödyllinen visualisointi on pintojen normaalit. Jos ne ovat pielessä, valaistuslaskennan tulokset voivat olla mitä sattuu. Normaaleissa on kolme komponenttia ja meillä on käytössä kolme värikanavaa – yksi kanava per komponentti. Koordinaattien ja värikanavien arvoalueet ovat hieman erilaiset, mutta tämä ratkeaa pienellä skaalausoperaatiolla.

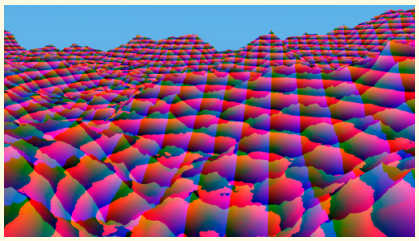
```
in vec3 normal;
out vec4 frag_color;
void main()
{
frag_color = vec3(normal * 0.5 + 0.5, 1.0);
}
```



Tässä kuvassa on visualisoitu pinnan normaalit kameran koordinaattiavaruudessa. Punertavilla alueilla X-komponentin arvo on nollaa suurempi ja Y-komponentin nollaa pienempi, eli ne osoittavat oikealle alas. Vaalean turkoosilla alueilla on paljon vihreää ja sinistä, joten suunta on ylös ja kohti kameraa.

Toisinaan on tarpeen visualisoida jonkin muuttujan arvon muutoksia suurella alueella. Yritys skaalata arvot lineaarisesti johonkin värikomponenttiin hävittää yksityiskohdat. Tällöin GLSL:n fract-funktiosta on hyötyä. Yhdessä sopivan skaalauksen kanssa sillä saa näppärästi näkyviin arvon isoviivat, kuten vaikkapa fraktaalimaaston korkeuskäyrät.

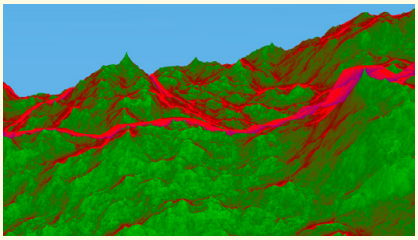
```
vec3 coord = trace_terrain(start_coord, obj_look_dir);
vec3 scaled = coord/vec3(32.0, 32.0, 16.0);
frag_color=vec4(fract(scaled.zyx)*vec3(1.0, 0.5, 0.8), 1.0);
```



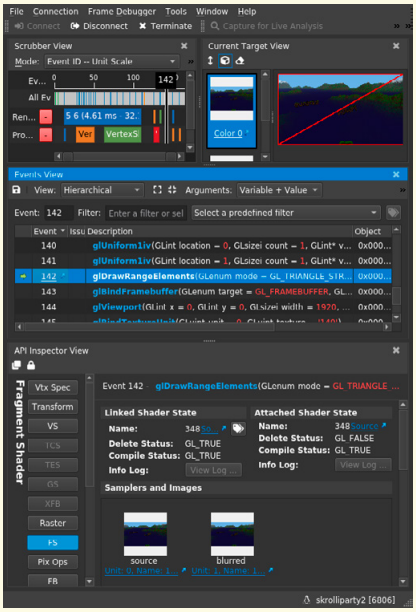
Punaiseen värikanavaan on visualisoitu korkeusarvot, jolloin etualalla oleva syvänne näkyy sisäkkäisinä renkaina. Vihreä ja sininen värikanava puolestaan sisältävät XY-koordinaatit, jolloin koordinaattiakselien suuntaiset korkeusprofiilit tulevat esiin.

Visualisoitavan arvon ei tarvitse olla suoraan tekemisissä renderöinnin lopputuloksen kanssa. Se voi liittyä myös esimerkiksi suorituskykyyn. Seuraavassa kuvassa maaston väri muuttuu vihreästä punaiseen ja lopulta violettiin, kun säteenseurannassa otettujen askelten lukumäärä kasvaa.

```
if(i<40)
return vec3((i-16)/48.0, 1.0-(i-16)/24.0, 0.0);
else if(i<64)
return vec3(0.5+(i-40)/48.0, 0.0, 0.0);
else
return vec3(64.0/i, 0.0, 1.0-64.0/i);
```



On myös olemassa erityisesti grafiikkakoodin vianetsintään tarkoitettuja ohjelmia. Tällainen on esimerkiksi NVIDIA Nsight Graphics. Sillä voi kaapata suoritettavasta ohjelmasta yksittäisen ruudun ja nähdä, mitä komentoja sen piirtämiseen käytettiin. Piirtokomentoja voi kelata eteen- ja taaksepäin nähdäkseen niiden vaikutuksen ruudun sisältöön. Shadereita voi jopa muokata lennossa. Tästä on suurta apua visualisoinnissa, koska ohjelmaa ei tarvitse kääntää tai edes käynnistää uudelleen jokaisen muutoksen jälkeen.



Tässä oli vain muutama esimerkki siitä, minkälaisia visualisointeja on mahdollista käyttää. Päästä mielikuvituksesi valloilleen ja keksi juuri omaan grafiikkaongelmaasi sopiva visualisointi!

Teksti ja kuvat:
Jarno Niklas Alanko

KOMPLEKSI-INTEGRAALIT
Uusi ratkaisu rekursioon

Fibonaccin jonoa 1,1,2,3,5,8,13… on helppo laskea perinteisellä tyylillä, mutta miksi tehdä asiat aina helpoimman kautta?

Fibonaccin lukujono on monen ensimmäinen kokemus rekursiosta matematiikassa. Matematiikassa rekursiivinen funktio on sellainen, jonka arvo pisteessä riippuu saman funktion arvosta edellisissä pisteissä – samaan tapaan kuin ohjelmoinnissa. Fibonaccin jonon kaksi ensimmäistä lukua ovat ykkösiä[1], ja seuraavan luvun saa aina lisäämällä kaksi edellistä yhteen. Se voidaan siis määritellä rekursiivisesti kaavalla $f(n) = f(n-1) + f(n-2)$ pohjatapauksin $f(1) = 1$ ja $f(2) = 1$.

Jonoa on helppo laskea suoraan määritelmästä, mutta on myös olemassa ratkaisu, joka perustuu kompleksilukuihin. Kompleksiluvut ovat lukupareja, joissa toista lukua kutsutaan reaaliosaksi, ja toista imaginaariosaksi. Nämä lukuparit voidaan ajatella pisteinä tasossa.

Fibonaccin jono löytyy eräästä kompleksifunktiosta, kun kompleksitason nollapisteen eli origon ympäri kierretään riittävän kaukaa. Kuulostaa ehkä hullulta, mutta antakaa kun selitän.

Matematiikkaa

Kompleksiluvuilla voi tehdä melkein mitä vain, mitä normaaleilla luvuilla voi tehdä. Niille on määritelty yhteen-, vähennys-, kerto- ja jakolaskut, joiden avulla päästään myös kiinni monimutkaisempiin käsitteisiin, kuten integraali. Kompleksi-integraali kulkee jotain annettua polkua pitkin kompleksitasossa ottaen eräänlaisen summan funktion arvoista polun varrella.

Kompleksifunktio on funktio, joka ottaa sisään kompleksiluvun ja palauttaa kompleksiluvun. Koodaamme Ficonaccin jonon seuraavaan kompleksifunktioon: $X(z) = 1 + z^{-1} + 2z^{-2} + 3z^{-3} + 5z^{-4}\ldots$ Tämä on siis ääretön sarja muuttujan z negatiivisia potensseja, joiden kertoimet muodostavat Fibonaccin jonon. Osoittautuu, että tällaisen negatiivisen potenssisarjan termin z^{-n} kertoimen voi laskea seuraavalla kaavalla:

$$\frac{1}{2\pi i}\oint_C X(z)z^{n-1}dz$$

Tässä kaavassa integraali on polkuintegraali kompleksitasossa, missä polku C on mikä vain sellainen polku, joka kiertää origon vastapäivään pysyen koko ajan funktion suppenemisalueessa. Kaavan täsmällinen laskeminen on vaikeaa, mutta sitä voidaan approksimoida numeerisesti tietokoneella mielivaltaisella tarkkuudella. Sillä voidaan siis laskea Fibonaccin jonon mikä tahansa luku laskematta erikseen jonon aiempia lukuja.

Ongelmana on vielä, että kaavan laskemiseen tarvitaan funktio $X(z)$, mutta siinä on äärettömän monta termiä, ja lisäksi sen muodostamiseen tarvitaan Fibonaccin jono, joka oli se asia, mitä tässä yritettiin alun perin laskea. Mutta ei hätää, funktio $X(z)$ voidaan kirjoittaa muotoon $X(z) = 1 / (1 - z^{-1} - z^{-2})$ pienellä matemaattisella tempulla. Nyt funktion voi laskea nopeasti äärellisellä määrällä laskutoimituksia, eikä sen laskemiseen tarvitse tietää Fibonaccin luvuista mitään.

[1] **Leonardo "Fibonacci" Pisano** *määritteli jonon alkamaan 1, 1 kirjassaan Liber Abaci. Nykyisin jono aloitetaan usein 0, 1. Noudatamme artikkelissa Fibonaccin omaa määritelmää – tätä tapaa käytetään muuten myös Turun voimalaitoksen savupiipussa olevassa* **Mario Merzin** *taideteoksessa. Fibonaccin alkuperäinen esimerkki laskee kaniparin ja sen jälkeläisten lisääntymistä.*

Teoria testiin

Oheisessa koodilistauksessa on C++-koodi, joka laskee Fibonacci-jonon 10 ensimmäistä lukua. Integrointi on toteutettu numeerisesti kulkemalla ympyräpolkua pitkin 0.0000001 yksikön pituisin askelin kerryttäen integraalin arvoa samalla. Tämä antaa arvion integraalin todellisesta arvosta, ja mitä pienempi askelpituus on, sitä parempi arvio. Jos kaikki menee kuten Strömsössä, saatu arvo parametrilla n on kompleksiluku, jonka reaaliosa on likimain *n*:s Fibonaccin luku ja imaginaariosa nolla. Ohjelma tulostaa seuraavaa:

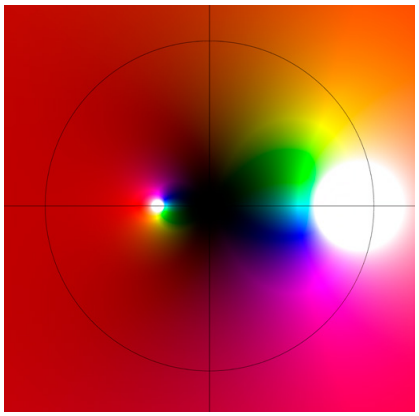
```
0 (1,3.14159e-07)
1 (1,3.14159e-07)
2 (2,6.28319e-07)
3 (3,9.42479e-07)
4 (5.00001,1.5708e-06)
5 (8.00001,2.51328e-06)
6 (13,4.08408e-06)
7 (21.0001,6.59736e-06)
8 (34.0001,1.06814e-05)
9 (55.0002,1.72788e-05)
```

Suluissa tulostetut lukuparit ovat kompleksilukuja, jossa parin ensimmäisen luku on reaaliosa toinen imaginaariosa. Koodi toimii kuten pitikin: Fibonaccin jono on luettavissa reaaliosista ja imaginaariosat hyvin lähellä nollaa.

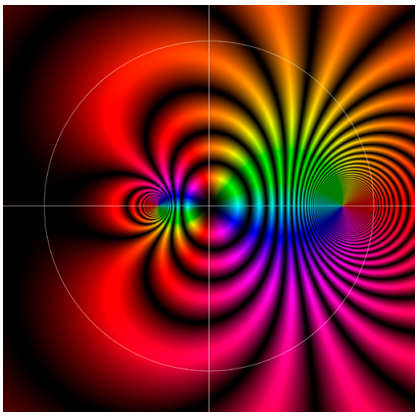
Onko tästä mitään hyötyä? Suoraan sanoen ei, koska suoraviivainen laskeminen olisi ollut nopeampaa, ja suuremmilla luvuilla liukulukujen epätarkkuus nousee ongelmaksi. On olemassa myös yksinkertaisempia kaavoja.

Menetelmä voidaan yleistää mille tahansa luonnollisille luvuille määritellylle rekursiiviselle funktiolle, joka voidaan kirjoittaa painotettuna summana funktion aiempia arvoja.

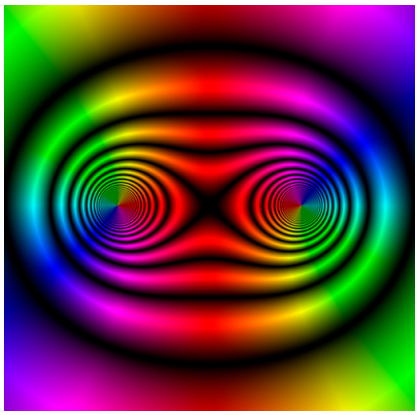
Jutun kuvituksena on esityksiä kompleksifunktioista. Kuvissa jokainen tason piste vastaa yhtä kompleksilukua ja pisteen väri kertoo funktion arvon siinä pisteessä. Pisteen värisävy vastaa kompleksiarvon kulmaa reaaliakselin kanssa ja kirkkaus liittyy kompleksiarvon itseisarvoon. Jutun funktio $X(z)$ on piirrettynä kuvissa 1 ja 2 niin, että ensimmäisessä pikselin kirkkaus on suoraan funktion $X(z)$ itseisarvo ja toisessa sellainen, että saman kirkkauden "vöissä" funktion itseisarvo pysyy samana. Integroinnissa ympyräpolun tulee kiertää pisteet, joissa funktio lähestyy ääretöntä. Nämä pisteet sijaitsevat kuvassa 1 jossain kohtaa kirkkaissa valkoisissa alueissa ja kuvassa 2 niissä kohdissa, joissa vyöt käyvät äärettömän tiheiksi. Lopuissa kuvissa esiintyy muita samankaltaisia funktiota.



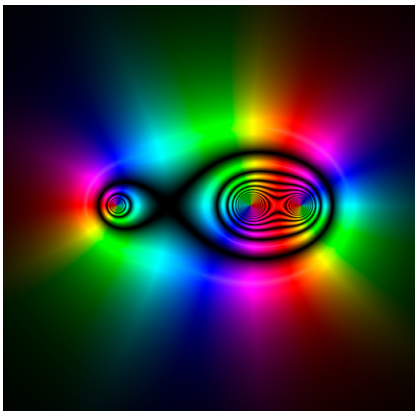
Kuva 1. Funktio $1 / (1 - z^1 - z^2)$.



Kuva 2. Funktion $1 / (1 - z^1 - z^2)$ tasa-arvokäyrät.

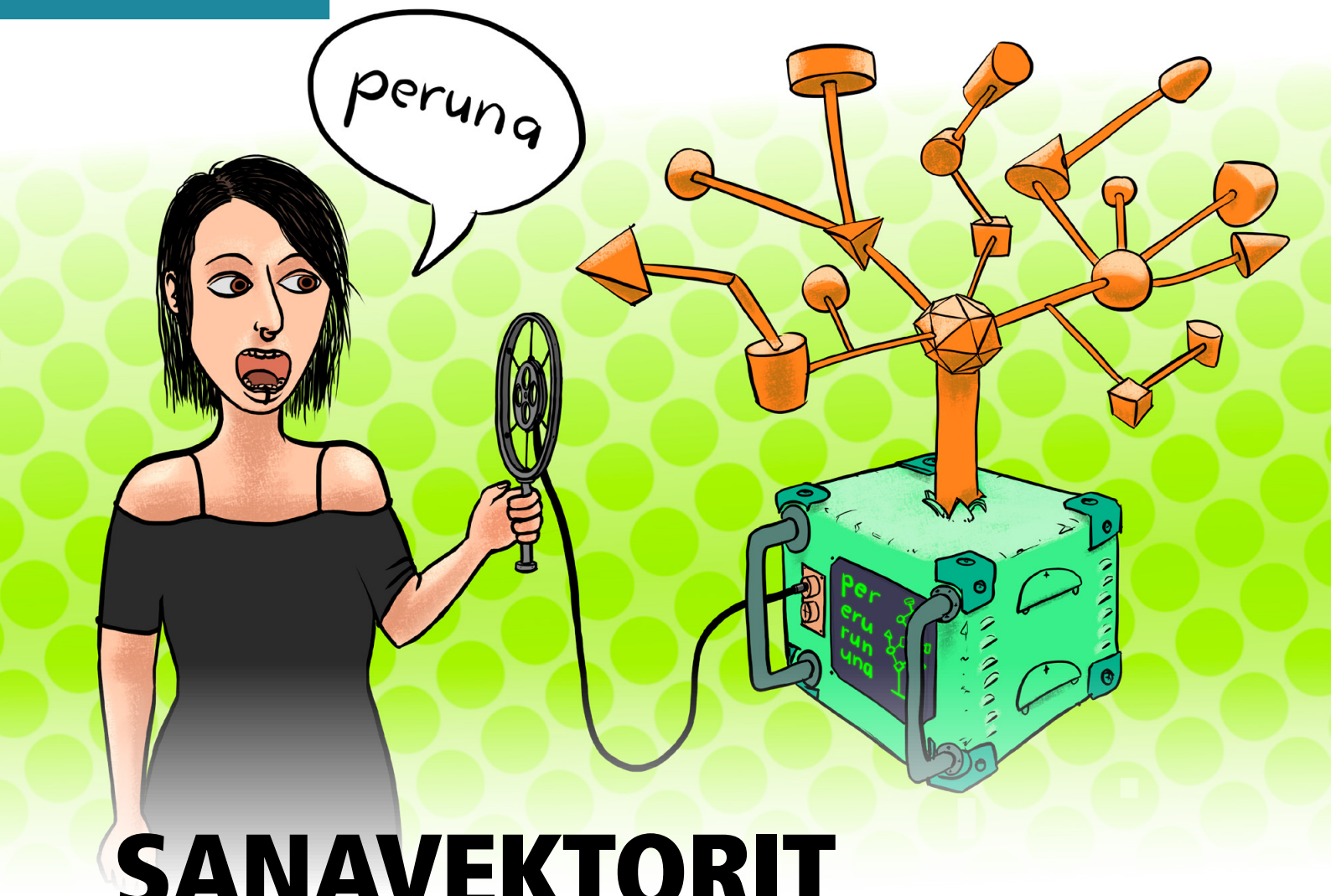


Kuva 3. Funktion $1 / (1 - z - z^2)$ tasa-arvokäyrät.



Kuva 4. Funktion $z^1 / (1 - z - z^2)$ tasa-arvokäyrät.

```
#include <complex>
#include <iostream>
using namespace std;
const double pi = 3.14159265359;

complex<double> integrate(complex<double> (*f)(complex<double>),
complex<double> (*path)(double), double a, double b, double step){
  complex<double> ans = 0;
  for(double t = a; t <= b; t += step){
    ans += step * (f(path(t)) * (path(t+step) - path(t)) / step);
  }
  return ans;
}

int n;
complex<double> fib(complex<double> z){
  return 1.0/(1.0 - 1.0/z - 1.0/(z*z)) * power(z,n - 1);
}
complex<double> circle(double t){
  return complex<double>(2,0)*exp(2*pi*complex<double>(0,1)*t);
}

// Lasketaan 10 ensimmäistä Fibonaccin lukua
int main(){
  for(n = 0; n < 10; n++){
    cout << n << " " << integrate(fib, circle, 0, 1, 0.0000001) *
(1.0/complex<double>(0,2*pi)) << endl;
  }
}
```



SANAVEKTORIT

Luonnollisen kielen käsittelyn salainen ase

Teksti: Ada-Maaria Hyvärinen Kuvat: Sakari Leppä, Jarno Niklas Alanko

Lapsena mietin, mikä on perunan vastakohta. Olisiko se porkkana vai jokin, mikä ei ole ruokaa lainkaan? Aikuiset sanoivat, ettei perunalla ole vastakohtaa. Tietääkö sanavektorimalli, mikä on perunan vastakohta?

Puhuvat ja kirjoittavat tietokoneohjelmat ovat kiinnostaneet ammattilaisia ja harrastajia koneiden synnystä asti. Skeptisimmät ajattelivat, ettei tietokone voi koskaan oppia ymmärtämään tai tuottamaan ihmismäistä kieltä. Viime aikoina laskentatehon kasvu on mahdollistanut tekoälysovellusten kehittymisen ennennäkemättömille tasoille myös automaattisen kielenkäsittelyn osalta. Skrollin edellisessä numerossa nähtiin tekoälyn avulla luotu lehtijuttu. Tässä jutussa taas paneudutaan suosittujen neuroverkkojen avulla tuotettujen sanaupotusvektorien kykyyn selviytyä kielellistä ymmärrystä mittaavista tehtävistä.

Koko ”kielellinen ymmärrys” on sinänsä vaikeasti määriteltävä käsite, ja siksi onkin vaikeaa sanoa, millaisilla kokeilla koneoppimismallin laatua kielellisissä tehtävissä on järkevää mitata. Sanavektorimallin laatua voi tutkia esimerkiksi katsomalla, osaako se kertoa, mitkä sanat tarkoittavat samaa tai edes liittyvät toisiinsa. Aikuiselle, suomea puhuvalle ihmiselle on esimerkiksi aika selvää, että *peruna, porkkana, mustajuuri, paprika* ja *kurkku* liittyvät toisiinsa paljon enemmän kuin vaikkapa *peruna* ja *sote-uudistus*. Mutta miten kielimalli voisi oppia tämän?

Konteksti on kaiken avain

Kielitieteilijä **J. R. Firth** esitti vuonna 1957 sanojen semantiikkaan eli merkitykseen liittyvän teorian: ”Sanan luonne määrittyy sen seuralaisten mukaan.” Tätä ajatusta kutsutaan *jakaumahypoteesiksi*, ja useimmat sanojen merkitystä käsittelevät koneoppimismallit perustuvat tähän.

Jos vaikka ystävämme kertoo meille *eilen illalla join liikaa Rieslingiä ja nyt on pää kipeä*, meidän ei tarvitse olla viiniasiantuntijoita ymmärtääksemme, että *Rieslingiä* viittaa alkoholijuomaan, ja että sanan tilalla voisi esiintyä vaikka *Karhua, cosmopolitania* tai *Koskenkorvaa*, tietysti ystävämme mieltymyksistä riippuen. Myös sanat *kahvia* tai *appelsiinimehua* sopisivat jotenkuten kontekstiin, vaikka emme ehkä tietäisi, miten pääkipu liittyy niihin. Vielä oudompaa olisi korvata sana vaikkapa sanoilla *suklaata, kivikkoa* tai *sarjakuvia*, vaikka sillä vaikka lause olisi edelleen kieliopillisesti oikein, ei konteksti sovi enää sanoihin. Huonoimmassa tapauksessa sanat eivät olisi kontekstissa oikein kieliopillisestikaan, kuten *hyppäisi*, tai edes suomen kielen sanoja, kuten *lampake*.

Sanan *Rieslingiä* voidaan siis ajatella olevan aika samanlainen kuin *Koskenkorvaa* ja aika erilainen kuin *hyppäisi*. Vaikka emme osaisi suomea, voisimme päätellä tämän tekemällä tilaston siitä, mitkä sanat esiintyvät usein yhdessä.

Tällaisen tilastollisen tiedon avulla voidaan opettaa koneoppimismalli, joka ”ymmärtää” sanojen merkityksen. Oikeasti malli ei tietenkään ymmärrä, mitä *juominen* tai *Riesling* ovat. Se oppii kuitenkin tunnistamaan niiden välisen yhteyden. Käytännön sovelluksiin se usein riittää.

Myös ihminen oppii sanoja kontekstin perusteella, mutta ihmisen kohdalla kontekstiin liittyy paljon muutakin kuin sanojen ympärillä olevat sanat. Ihminen liittää sanoja myös esimerkiksi sosiaalisiin suhteisiin ja aistikokemuksiin. *Rieslingistä* voisikin tulla mieleen vaikka viinin maku, hyvin sujunut illallistilaisuus tai entisen seurustelukumppanin mieltymykset. Sen sijaan tilastollisten mallien on opittava sanoihin liittyvä tieto vain toisten sanojen perusteella. Opetusaineiston onkin oltava riittävän kattavaa ja aiheeseen sopivaa, jotta opetus onnistuu riittävän hyvin.

Analogiat

Kielellistä ymmärrystä on siis ainakin samankaltaisten sanojen tunnistaminen ja erilaisten sanojen erottaminen toisistaan. Toinen tapa mitata ihmisen tai koneen kielellistä osaamista on testata taitoa *analogiaparien* tuottamiseen. Esimerkiksi *pieni* ja *suuri* ovat toistensa vastakohtia, joten niiden suhde on analoginen sanojen *hidas* ja *nopea* kanssa, jotka ovat myöskin toistensa vastakohtia.

Vastakohtaisuus liittyy sanan merkitykseen. Analogiapareja voi kuitenkin keksiä myös sanan muotoon liittyen: vaikkapa *kissa, kissat* on analoginen sanaparin *koira, koirat* kanssa, koska kummassakin on sanan yksikkö ja monikko.

Ensi silmäyksellä voisi ajatella, että tietokoneelle tällaiset kielioppisääntöjen mukaiset analogiat ovat paljon helpompia kuin merkitykseen liittyvät. Sanaan *kissa* tarvitsee vain lisätä loppuun *-t*, samoin sanaan *koira*, ja monikkomuoto on valmis. Tietokoneen voi myös luottaa osaavan taivuttaa harvinaisempia sanoja, kuten *susi, suden*: tällaisia sanamuotoja tuottavia ohjelmia voi tehdä ilman koneoppimista kattavien sääntökokoelmien avulla.

Vaikka myös tietopankkeja esimerkiksi vastakohdista on koottu, tuntuu intuitiivisesti vaikeammalta saada tietokone ymmärtämään, millainen on *lehmän* ja *vasikan* tai *äidin* ja *lapsen* suhde. Sellaisen ajattelisi vaativan tietoa reaalimaailmasta, lehmien ja ihmisten perhe-elämästä. Koneoppimisen avulla kielimalli voi kuitenkin pystyä yllättäviin suorituksiin. Sanat täytyy vain ensin muuttaa tietokoneelle helpompaan muotoon: vektoreiksi.

Miksi vektorit?

Sanat esitetään tietokoneessa normaalisti jonoina merkkejä. Merkkijonomuotoiset sanat ovat moneen tarkoitukseen hyviä: esimerkiksi englantia käsittelevä keskustelubotti voi reagoida asiakkaan tuottamaan merkkijonoon *no* ja tietää siitä keskeyttää toimintonsa. Suomenkielisessä versiossa vaaditaan joko botilta kykyä tunnistaa lista kieltosanoja (vaikkapa *ei*, *en* ja *emme*) tai kehittäjältä mielikuvitusta muotoilla kysymykset siten, että luonteva vastaus on aina pelkkä *ei*.

Sanavektoreihin perustuva ratkaisu sijoittaa kaikki kieltosanat lähekkäin abstraktiin sana-avaruuteen. Siten jos esimerkiksi käyttäjä sanoo botille *en*, voimme katsoa sana-avaruudesta, että se on lähellä sanaa *ei*, ja toivoa, että käyttäjä yrittää sanoa samaa kuin joku, joka sanoisi *ei*. Tietoa sanojen samankaltaisuudesta voidaan hyödyntää samalla tavalla vaikkapa dokumenttien luokittelussa, tiedonhaussa, konekäännämisessä tai mitä kielimallin avulla sitten yrittääkään tehdä.

Sanat voidaan esittää vektoreina monenlaisella tavalla, ja periaatteessa sanavektorilla voitaisiin tarkoittaa mitä tahansa sanoja kuvaavaa vektoria. Tässä tekstissä keskitytään kuitenkin sanaupotusvektoreihin eli word embedding -vektoreihin, jotka ovat nulotteisia liukulukuvektoreita. Sanojen esittäminen liukulukuvektoreina toimii koneoppimisen kanssa hyvin yhteen, koska useimmat koneoppimismenetelmät on suunniteltu käsittelemään nimenomaan liukulukuvektoreita.

Sanat säkkiin

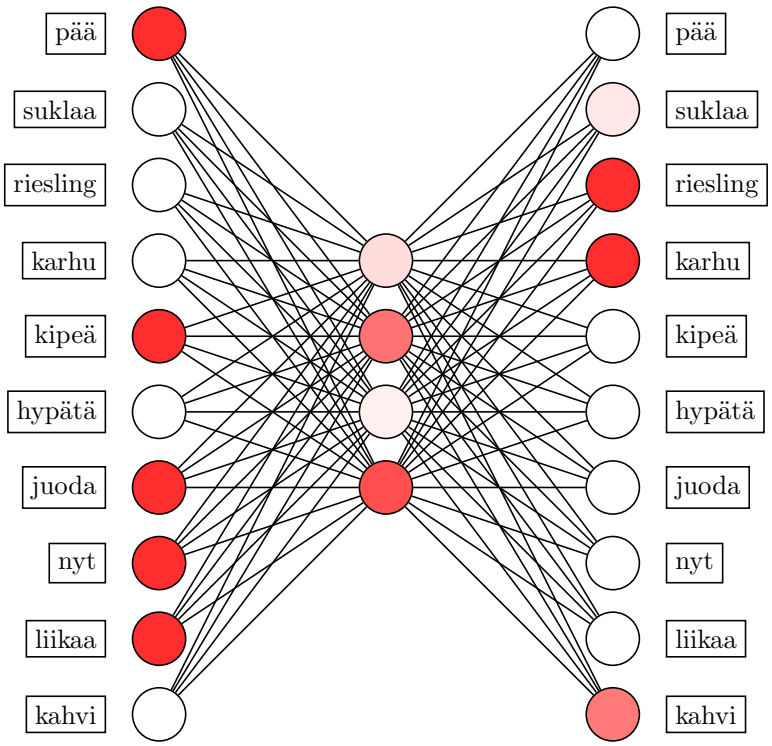
Nykypäivänä tunnetuin sanavektorimalli on Googlen tutkijoiden kehittämä *word2vec*. Se opetetaan niin sanotun valetehtävän avulla. Oikeasti ollaan kiinnostuneita mahdollisuudesta kuvata sanoja sellaisilla vektoreilla, että samankaltaisten sanojen vektorit ovat toisiaan lähellä ja erilaisten sanojen vektorit ovat toisistaan kaukana. Verkkoa ei kuitenkaan opeteta suoraan toteuttamaan tämä tavoite. Sen sijaan se koulutetaan arvaamaan sana kontekstin perusteella.

Opetusdata voidaan luoda kahdella tavalla: joko skip-gram-mallilla tai CBOW-mallilla. Tutustutaan kuitenkin ensin CBOW-malliin, eli Continuous Bag Of Words -malliin. Tässä syötteenä on tekstinpätkiä, joista puuttuu keskeltä sana, esimerkiksi *anna minulle armahduksesi ___ minulle Rafaelin enkeli*. Valetehtävänä on arvata, mikä sana kontekstista puuttuu. Tässä tapauksessa vastaus on sana *ole*.

Arvaaminen tehdään neuroverkolla, jossa on syötekerros, yksi piilokerros ja ulostulokerros. Jokainen syötekerroksen neuroni esittää yhtä sanaston sanaa. Siis jos opetusdatassa on 10 000 erilaista sanaa, syötekerroksessa on 10 000 neuronia. Konteksti syötetään verkkoon antamalla kontekstin sanoja vastaaville neuroneille arvoksi nollasta poikkeava arvo ja muille nolla.

Piilokerroksen koko vastaa haluttua sanavektorien ulottuvuutta. Tavanomaisesti ulottuvuus on joitakin satoja. Ulostulokerroksessa on taas yksi neuroni jokaista sanaston sanaa kohti, ja neuronin aktivaatioarvo tulkitaan todennäköisyytenä sille, että annetun kontekstin väliin kuuluu tämä sana.

Verkkoa opetetaan syöttämällä sille suuri joukko konteksti-välisana-pareja ja säätämällä verkon painokertoimia vastavirta-algoritmilla (*backpropagation*) hiljalleen oike-



CBOW-mallin opetus. Kuvassa verkkoon syötetään sanat ”pää”, ”kipeä”, ”juoda”, ”nyt” ja ”liikaa”. Punaiset värit kuvaavat neuroneiden aktivaatioita: mitä kirkkaampi väri, sitä vahvempi aktivaatio. Oikealla olevassa ulostulokerroksessa syttyvät ne neuronit, joiden edustamat sanat esiintyvät annettujen sanojen kanssa usein yhdessä. Sanavektorit voidaan lukea verkosta aktivoimalla ainoastaan haluttu sana syötekerroksessa ja lukemalla piilokerroksen aktivaatiot.

aan suuntaan. Kun verkkoa on opetettu tarpeeksi, sanavektorit voidaan lukea siitä syöttämällä sille sanat yksitellen ja lukemalla piilokerroksen aktivaatioarvot. Valetehtävän oppinutta verkkoa ei tämän jälkeen tarvita enää.

Mitä suurempi piilokerros verkolla on, sitä rikkaammin malli voi kuvata sanaston ominaisuuksia. Vektorien laatu ei kuitenkaan parane kasvattamalla piilokerrosta valtavan suureksi.

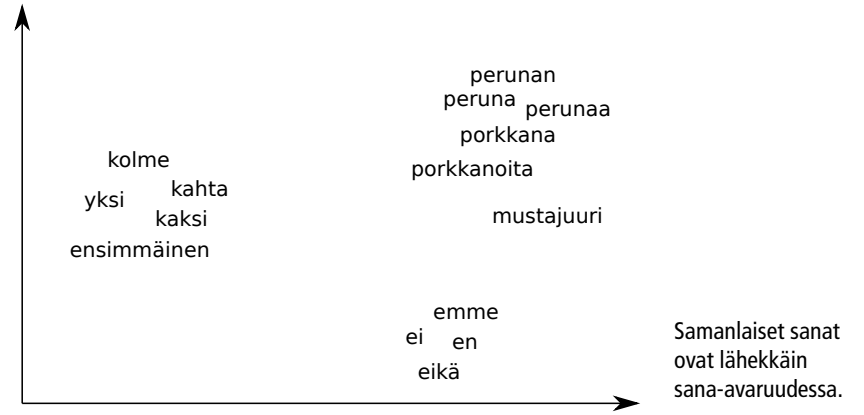
Skip-gram-malli toimii melkein samalla tavalla kuin CBOW-malli. Kun CBOW-mallissa arvataan kontekstiikkunasta puuttuva sana, tehdään skip-gram-mallissa päinvastoin. Siinä malli opetetaan arvaamaan, mitkä sanat kuuluvat välisanan konteksti-ikkunaan. Edellisen esimerkin tilanteessa verkon pitäisi siis tietää, että *anna, minulle, armahduksesi, minulle, Rafaelin* ja *enkeli* kuuluvat sanan *ole* kontekstiin. Myös skip-gram-malli opetetaan vastavirta-algoritmilla, ja siitäkin poimitaan sanavektoreiksi piilokerroksen aktivaatiot.

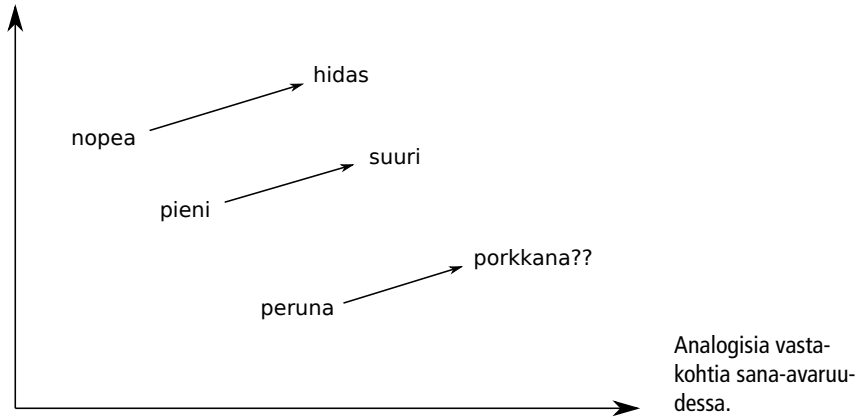
Yllättäviä sovelluksia

Nyt toivon mukaan rakentamamme vektorimalli sisältää semanttista tietoa kielestä. Voisimme esimerkiksi yrittää löytää kaikki sanaston kieltosanat etsimällä vektorit, jotka ovat lähellä *ei*-sanaa. Tämä olisi hyödyllistä esimerkiksi chatbot-projektissamme. Sanavektoreiden samankaltaisuutta mitataan niiden kulman kosinilla.

Kontekstitiedon avulla opetetut word2vec-vektorit sisältävät kuitenkin paljon muutakin tietoa kuin vain sen, mitä sanaa mikäkin vektori kuvaa ja mitkä sanat ovat samanlaisia. Yllättäen osoittautuu, että myös word2vec-vektorien komponentittaisella yhteen- ja vähennyslaskulla on tulkinta, joka liittyy sanojen merkityksiin.

Merkitään sanaa vastaavaa vektoria merkinnällä v(sana). Vektoreilla voidaan tehdä esimerkiksi seuraavanlaisia laskutoimituksia: v(Berliini) - v(Saksa) + v(Ranska). Laskun tulokseksi tulee likimain v(Pariisi). Pariisi on siis vähän kuin Berliini, josta on poistettu Saksa ja lisätty tilalle Ranska. Ei ole mitenkään ilmeistä miksi näin tapahtuu, mutta tälle on olemassa monimutkainen perustelu, joka liittyy word2vec-mallin optimoitavan funktion matemaattiseen muotoon.





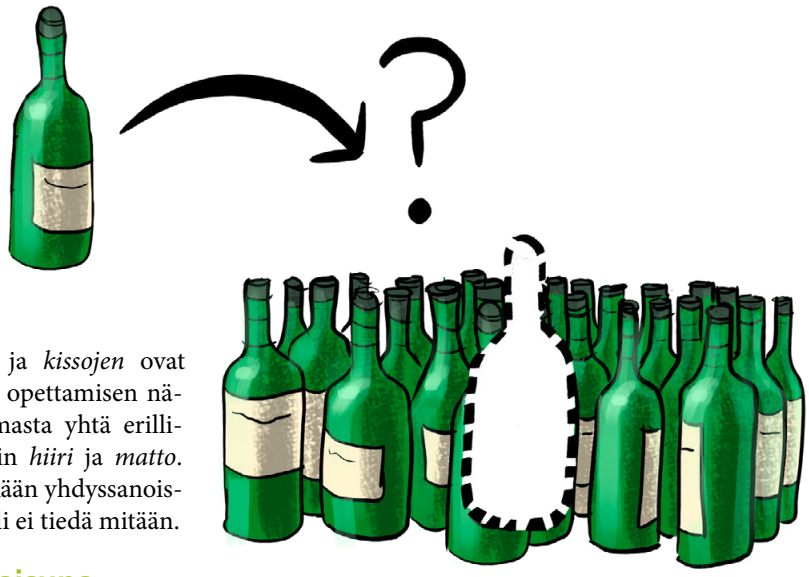
Suomen kielen haasteet

Suomalaiset pitävät suomen kieltä todella vaikeana kielenä niin ulkomaalaisille kuin tietokoneillekin, eikä tämä käsitys ole täysin väärä. Useimmat sanatasolla toimivat mallit toimivat suomen kielellä huonommin kuin englannilla, koska suomen kieli on agglutinoiva ja morfologisesti rikas, eli suomen kielessä paljon informaatiota välitetään taivuttamalla sanoja, ja sen takia suomessa on paljon erilaisia sananmuotoja.

Verkon opettamiseksi pitää päättää, minkä kokoista sanastoa hyödynnetään. Riippuu paljon tehtävästä, pitäisikö sanastoon kuulua kymmenentuhatta, satatuhatta vai miljoona sanaa vai kenties kaikki lähtöaineistossa (esimerkiksi Wikipediassa) esiintyvät sanat. Tavallaan oli huijausta puhua kymmenestätuhannesta sanasta, koska se voi johtaa ajattelemaan, että nuo kymmenentuhatta sanaa esiintyvät sanastossa perusmuodossa niin kuin sanakirjan hakusanat. Jos kuitenkin poimitaan jostakin oikeasta tekstistä kymmenentuhatta yleisintä sanaa, niin englannin- kuin suomenkielisellä aineistolla osa sanoista esiintyy taipuneessa muodossa. Tästä kieliriippumaton koneoppimismallimme ei kuitenkaan tiedä mitään, vaan se tallentaa sanat kiltisti siinä muodossa, jossa ne esiintyvät tekstissä.

Usein sanastoon valitaan kaikki sanat, jotka esiintyvät opetusaineistossa tarpeeksi usein, esimerkiksi vähintään viisi kertaa. Opetuksen jälkeen sanasto on valmis. Malli ei osaa tuottaa vektoreita ennennäkemättömille sanoille.

Sanastoon kuulumattomien sanojen kanssa toimimattomuus voi muodostua ongelmaksi millä tahansa kielellä. Taivuttavien kielten kuten suomen kanssa herää myös toinen ongelma: sanojen erillisyys. Malli käsittelee jokaista sanaa itsenäisenä, eikä välitä siitä, muistuttaako se merkeiltään jotakin muuta sanaa. Esimerkiksi



kissan ja *kissojen* ovat mallin opettamisen näkökulmasta yhtä erillisiä kuin *hiiri* ja *matto*. Myöskään yhdyssanoista malli ei tiedä mitään.

Ratkaisuna alimerkkijonot

Ongelmalle löytyy onneksi ratkaisuja, kuten viime aikoina suosiota saavuttanut word2vecin laajennos fastText. Facebookin kehittämä fastText-malli toimii muuten samalla tavalla kuin word2vec, mutta mallille opetusaineistona annettavat sanat esitetään eri muodossa. Nämä sanavektorit muodostetaan sanaa vastaavasta vektorista sekä sen alimerkkijonoja vastaavista vektoreista. Alimerkkijonoilla eli merkki-n-grammeilla tarkoitetaan tässä yhteydessä n:n pituisia sanan osia. Jos otettaisiin vaikkapa sana *kissa* ja kaikki sen 3-grammit, saataisiin joukko {*kissa, kis, iss, ssa*}. Todellisuudessa sanan alku ja loppu on vielä merkitty erikoismerkeillä < ja >, jolloin joukko on hiukan laajempi: {<*kissa*>, <*ki, kis, iss, ssa, sa*>}. FastText-mallissa on oletuksena otettu mukaan kaikki n-grammit, joissa n on välillä 3–6. Kielestä ja mallin tarkoituksesta kuitenkin riippuu, mitkä n-grammit kannattaa ottaa mukaan.

Koska fastText-mallissa sanojen sisältämät merkkijonot vaikuttavat niiden vektoriesitykseen, muistuttavat samoja merkkijonoja sisältävien sanojen vektorit toisiaan enemmän kuin merkkijonojen sisällölle sokeassa word2vec-mallissa. Tämä sinänsä hyvin yksinkertainen kikka auttaa saman sanan eri taivutusmuotoja kuvautumaan lähemmäs toisiaan. Se myös helpottaa merkitysanalogioiden (kuten aiemmin nähdyt maat ja pääkaupungit) ja muotoon liittyvien analogioiden (kuten vaikkapa *iso, isoin* ja *kylmä, kylmin*) löytymistä sanavaruudesta.

Myöskään fastText-malli ei tiedä mitään sanojen taivuttamisesta, vaan se on kieliriippumaton ja perustuu pelkästään peräkkäisiin merkkeihin. Yksinkertaisuudestaan huolimatta alimerkkijonojen käyttö parantaa mallin kykyä tuottaa laadukkaita suomenkielisiä vektoreita. Vektoreita tuottaessa kannattaa kuitenkin valita tarpeeksi suuri piilokerroksen koko, jotta malli pysyy tarpeeksi rikkaana tallentamaan sekä alimerkkijonojen tuoman tiedon että word2vec-mallin henkisesti kontekstitiedon. Pieniulotteisilla vektoreilla on vaarana, että malli painottuu liikaa alimerkkijonojen osalta. Silloin se alkaa nähdä yhteyksiä vaikkapa sanojen *joulu* ja *koulu* välillä. Niillä on toki pitkä yhteinen alimerkkijono, ja joissakin kielissä sanoja taivutetaan ja muodostetaan nimenomaan sanan alkuosaa muuttamalla. Malli ei voi tietää, ettei suomi kuulu näihin kieliin. Kontekstitiedon avulla malli kuitenkin oppii, etteivät nämä sanat yleensä esiinny kovin samanlaisissa yhteyksissä, ja osaa siksi pitää niitä erillisinä.

Entä se perunan vastakohta

Nyt kun tiedämme, mikä fastText-malli on ja miksi se pystyy vastaamaan analogiakyselyihimme, voimme tutkia, löytäisikö se ratkaisun lapsuusaikojen kysymykseen perunan vastakohdasta. Olisihan hienoa, jos tekoäly pystyisi todistetusti ratkaisemaan tämänkin ihmiselle suorastaan mahdottoman ongelman!

Ensin tarvitsemme esimerkin vastakohtaisuusparista, esimerkiksi aikuinen–lapsi. Nyt voimme pyytää mallia täyttämään muuttujan x seuraavaan

lauseeseen: ”sanan *lapsi* suhde sanaan *aikuinen* on sama kuin sanan *peruna* suhde sanaan *x*”. Tämä onnistuu vähentämällä aikuisen vektorista lapsen vektori ja lisäämällä perunan vektori, minkä jälkeen tuloksen pitäisi olla perunan vastakohta. Kokeillaan vastakohtien löytämistä skip-gram-mallilla, jossa vektorin pituus on 300 ja opetukseen on käytetty 3-, 4-, 5- ja 6-grammeja. Nämä ovat fastText-kehittäjien tavallisesti käyttämät parametrit, joilla opetettuja valmiita malleja voi esimerkiksi ladata kirjaston sivuilta. Taulukossa 1 on listattu muutamalle vastakohtaparille kaksi mallin löytämää perunan parasta vastakohtaa:

Vastakohtapari	Perunan vastakohta
aikuinen, lapsi	perunat, vientiperuna
pieni, suuri	puikulaperuna, perunat
pitkä, lyhyt	perunaa, perunaan
heikko, vahva	perunan, perunaan

Taulukko 1.

Ei näytä lupaavalta. Vastakohtapariyritykset osuvat erilaisiin perunan taivutusmuotoihin tai perunatyyppeihin. Entä oma ehdotukseni perunan vastakohdaksi, *porkkana*? Vastakohtaparilla *heikko, vahva* malli tarjoaa *porkkana*-sanalle vastakohdiksi *porkkanana* ja *porkkanan.*

Ehkä alimerkkijonot haittaavat kyselyn toimimista. Selvästi malli tietää, että perunaan liittyviä asioita ovat sanan eri taivutusmuodot. Kuitenkaan vastakohtaisuutta ei näy mallin antamissa tuloksissa.

Kokeillaan mallilla, joka on opetettu perinteiseen word-2vec-tyyliin ilman alimerkkijonoja. Jätetään siis n-grammit pois ja opetetaan samalla aineistolla 300-ulotteisia vektoreita tuottava skip-gram-malli.

Toisella mallilla tulokset ovat erilaisia, mutta eivät välttämättä sen hyödyllisempiä. Tulokset on listattu taulukossa 2.

Vastakohtapari	Perunan vastakohta
heikko, vahva	juurekset, rypsi
pieni, suuri	juurekset, mustaselja
pitkä, lyhyt	jamssi, palkokasvit

Taulukko 2.

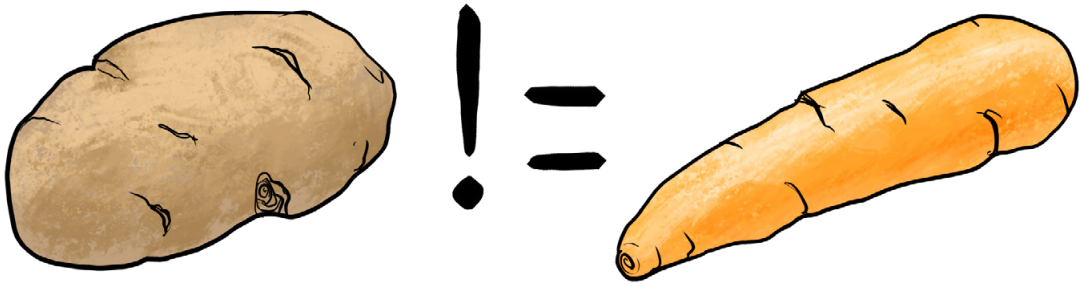
Tällä kertaa perunan vastakohtaa lähinnä on siis perunan taivutusmuotojen sijaan muita kasviksia. Kun etsitään vastakohtaa sanalle *porkkana* vastakohtaparina *pitkä, lyhyt* saadaan *munakoiso* ja *tärkkelysperuna*. Ehkä en ollutkaan lapsena ihan väärässä perunan ja porkkanan vastakohtaisuudesta?

Tyypillistä analogiakyselyille on osua johonkin hakusanoista tai niitä muistuttaviin sanoihin, kuten synonyymeihin tai sananmuotoihin. Täytyy kenties myöntää, ettei perunalle löydy luonnollista vastakohtaa, tai ainakaan perunan ja sen vastakohdan suhde eivät muistuta kahden adjektiivin vastakohtaisuutta. Epätoivoisesta kyselystämme huomataan ainakin alimerkkijonojen merkitys: kun ne ovat käytössä, perunan lähelle kuvautuu enemmän sanan taivutusmuotoja. Ilman alimerkkijonoja perunan luota löytyy muita syötäviä ja kenties jopa samanlaisissa ruuissa käytettäviä kasveja.

Sanavektorit eivät siis ole ihmistä viisaampia, mutta niitä hyödyntävillä malleilla tavoitetaan kenties jotakin siitä, miten ihminenkin hahmottaa kielen. Suomen kielen sananmuotojen omaksumiseen yksinkertainen ja kieliriippumaton kikka, n-grammien käyttö, auttaa jo selvästi. Riippuu sanavektorien käyttötarkoituksesta, kuinka tärkeää on saada saman sanan eri muodot lähelle toisiaan. Toisaalta n-grammeja hyödyntävä fastText on word2vec-menetelmää käytännöllisempi myös sanaston joustavuuden kannalta. Alimerkkijonojen avulla voidaan laskea vektoriesitys mille tahansa sanalle, vaikka sitä ei esiintyisi opetusaineistossa lainkaan. Monia erilaisia sananmuotoja sisältävälle suomelle tästä on hyötyä, mutta ominaisuus on kätevä myös vaikkapa yhdyssanojen tai uudissanojen kannalta.

Omien sanavektorimallien opettaminen on helppoa ja nopeaa saatavilla olevien kirjastojen ja avoimien aineistojen avulla. Jos siis haluaa tehdä luonnolliseen kieleen liittyvää koneoppimista, kannattaa harkita, haluaisiko parantaa mallia opettamalla sitä tehtävään sopivalla aineistolla ja valitsemalla tehtävän mukaan, haluaako käyttää alimerkkijonoja vai ei.

Word2vec-malli toimii käytännössä erittäin hyvin, mikä on hämmentävää ottaen huomioon miten yksinkertaisiin periaatteisiin se perustuu. Ehkä liiankin hyvin. Tutkijat etsivät edelleen teoreettisia selityksiä mallin toimivuudelle.



Janne Sirén

Koodin sisennys
Tabulaattori vs. välilyönti

Jarno Niklas Alanko

Tabulaattori

Lähdekoodi tulee ehdottomasti sisentää tabulaattorimerkillä (ASCII 9, horizontal tab). Välilyönnillä sisentäminen kiihdyttää ilmastonmuutosta, edistää suljettuja ohjelmistoratkaisuja ja on haitallista kansanterveydelle.

Tabulaattorilla sisentäminen on energiatehokkain sisennystapa. Miljoona lähdekoodiriviä sisältää noin kolme miljoonaa sisennystä. Tabulaattorimerkeillä tilavaade pysyy samana. Välilyönneillä sisennettynä sisennyksen merkkimäärä kasvaa noin 15 miljoonaan, mikä vaatii enemmän tallennustilaa ja prosessoriaikaa – energiaa.

Tabulaattori on sisennyksen avoin lähdekoodi: standardinmukainen, vailla toimittajalukkoa (vendor lock-in) ja rajattomalla muokattavuudella. Sisentäminen tabulaattorilla ei ole riippuvaista alkuperäisen kehittäjän valinnoista toimiakseen kaikkialla ja tehokkaasti. Tabulaattorin esittämää sisennyssyvyyttä on myös triviaalia muuttaa, joten se skaalautuu tarpeisiin, joita alkuperäinen kehittäjä ei osannut tai halunnut huomioida.

Tutkimusten mukaan[1] yksi kolmesta suomalaisesta ohjelmoijasta onkin joutunut sairaalahoitoon muokattuaan välilyönneillä sisennettyä koodia alkuperäisen kehitysympäristön ulkopuolella.

[1] Lähde: Hevosmiesten tietotoimisto

Välilyönti

Koodi sisennetään välilyönneillä. Tärkein syy tähän on, että välilyönti näyttää samalta kaikille, toisin kuin tabi, jonka leveys riippuu tekstieditorin asetuksista. Koodin kuuluu näyttää siltä, kuin sen kirjoittaja tarkoitti.

Välilyöntejä voi käyttää myös koodirivien linjaamiseen esimerkiksi siten, että peräkkäisten rivien yhtäsuuruusmerkit osuvat kohdilleen. Jos sisennyksesi on tehty tabeilla, joudut sekoittamaan tabeja ja välilyöntejä tässä kohtaa. Nyt sinulla on kahdenlaisia whitespace-merkkejä peräkkäin samalla rivillä, mikä yksinkertaisesti tuntuu pahalta. Lisäksi tässä on vaara, että joku erehtyy linjaamaan tabeilla, jolloin linjaus menee rikki, kun tabin leveys muuttuu. Jos virhe on mahdollista tehdä, joku tekee sen ennen pitkää.

Kun sisennyksen leveys on yksikäsitteinen, voidaan asettaa vaatimukseksi rajata koodirivin leveys esimerkiksi 80 merkkiin, jolloin sitä voi katsella mukavasti standardilevyisestä terminaalista. Ennen kaikkea, välilyöntien käyttö on merkki hyvästä mausta. Stackoverflown kyselyssä[2] välilyöntien käyttö korreloi selvästi positiivisesti vuosipalkan kanssa. Efekti on riippumaton esimerkiksi ohjelmointikielestä ja sovellusalueesta. Teoriani mukaan välilyöntien käyttö kielii hyvästä tyylitajusta, joka näkyy myös palkassa ennen pitkää.

[2] stackoverflow.blog/2017/06/15/developers-use-spaces-make-money-use-tabs

Vastaus Niklakselle

Välilyöntisisennykset tarjoavat näennäistä joustavuutta koodiesityksen kovakoodaamiseen. Tämän apukepin vuoksi sisennysmerkkien sekakäyttäjät ovat toki ongelma.

Puhdasoppinen tabulatoristi ei sekaannu väliyönteihin, eikä tee oletuksia lukijan ympäristöstä. Tabulaattori on kuvauskieltä, jonka esitysmitta on täysin dynaaminen. Tabulaattorin perusteella moderni editori esittää koodin kulloinkin sopivimmassa muodossa – samalla vaikka tasaten metodin parametrit tai ne yhtäsuuruusmerkit.

Emme tee ohjelmiakaan enää vain Windows-työpöydälle, miksi koodaisimme 80 merkin terminaalille.

Vastaus Jannelle

Tervetuloa nykyaikaan, Janne. Tallennustilan säästö oli ehkä relevantti argumentti Commodore 64:n aikoihin, mutta 15 miljoonaa merkkiä on 15 megatavua, jotka eivät tunnu missään nykyaikana. Yksi pakkaamaton kuvatiedosto voi olla isompi.

Koodin sisennyssyvyyttä ei kuulu muuttaa, koska tabin leveys vaikuttaa siihen, kuinka koodi kannattaa rivittää. Kahden levyisillä tabeilla kauniisti muotoiltu koodi saattaa näyttää huomattavasti huonommalta kahdeksan levyisillä tabeilla. Jos eri osat koodista on kirjoitettu eri pituisia sisennyksiä ajatellen, kokonaisuutena koodi ei näytä hyvältä millään leveydellä.

Valokuvat: Janne Sirén, Tuukka Norri
Grafiikat: Janne Sirén

Teksti: Jukka O. Kauppinen
Kuvat: Laitevalmistajat

Mukaan matkaan ja mökille

KESÄN TEKNOLELUT

Kesäkuuma, lelujen huuma. Kun kelit kohentuvat ja päivät pidentyvät, uusien teknolelujen houkutukseen on helppo sortua. Skrolli antaa viime hetken laitevinkit äkkilähdölle.

Uuden kesälelun voi löytää mistä vain. Ehkä se hyppää silmille kaupan hyllyjen ohi kävellessä. Ehkä se on varsinainen villitys tai kätevä vempain, josta voisi teoriassa olla joskus hyötyäkin. Tekemistä mökille tai tuikitärkeä varuste matkalle. Mitä löysimmekään tälle kesälle?

Hintalaatua Lidlistä

Yksi erikoisemmista silmille hyppimisistä tapahtui Lidlissä, jonka hyllylle ilmestyi jo viime kesänä 27 euron *Quadcopter*, nelikopteri. Ai alle 30 euron lentolaite? Kyllä, kiitos.

Halpislenturin laadukkuus yllätti. Vaikka ohjain on kiinaa, ohjaustikut ovat verrattain tarkat. Kelvollisen gyron ansiosta laite on vakaa ja sutjakka lennätettävä. 2,4 gigahertsin taajuudella toimiva nelikanavainen ohjaus toimii oivallisesti, ja jos häröjä sattuukin, roottorit ovat hyvin suojattuja.

Halpa Lidl-kopteri on hintaansa nähden asiallinen ja ihmeen hauska ja helppo lennätettävä. Ainoa selvä miinus tulee USB:llä ladattavan akun pikkuruisuudesta, sillä siinä riittää virtaa vain 4–6 minuutin mittaisiin lentoihin. Akku on kuitenkin kätevästi omassa avattavassa kotelossaan, joten askartelija voi vaihtaa sen suurempaan.

Tämänkesäinen suurempikokoinen, noin 35 euron nelikopteri vaikuttaa olevan samalta valmistajalta. Laatikon mukaan se lentäisi jo kahdeksan minuuttia kertalatauksella. Mieli teki, mutta eiköhän yksi lelunelikopteri riitä tähän taloon.

Isommat sähkölelut näyttävät olevan tämän kesän suuri muotivillitys, sillä sähköpotkulaudat ovat rantautuneet vauhdilla sekä koteihin että ainakin Helsingin ja Tampereen kaduille vuokrattavina kiitäjinä. Mutta ei liene lainkaan liioiteltua väittää, että jos kaupassa on vierekkäin 150 ja 500 euron laudat, niin niiden materiaaleissa, tekniikassa ja jopa akkujen paloherkkyydessä voi olla hieman eroa.

Xiaomi Mi Electric Scooter M365 asettuu mittarin 500 euron päähän. 12-kiloinen e-lauta taittuu kokoon ja mahtuu auton takakonttiin, mutta kulkee taitetusta tangosta kannettuna mukana myös bussissa tai metrossa. Ja kun laudalla kiitää, niin johan siinä lähes liitää, kyyti on pehmeää ja mukavaa. Ymmärrän hyvin miksi lautojen suosio on kasvanut hurjaa vauhtia, sillä kyllähän nämä ovat käytännöllisiä kaupunkiliikenteeseen ja jopa työ- ja hupimatkojen yhdistelmäliikkumiseen. Samoin ymmärtää kritiikin niiden vaarallisuudesta, turhakkeisuudesta sekä vuokralautojen päättömästä pysäköinnistä.

Riskitekijätkin kannattaa muistaa. Kevyen liikenteen väylillä täytyy ymmärtää oma äänettömyytensä ja nopeutensa. Kypärä on tärkeä, sillä sähkölaudalla kaatuu helpommin ja dramaattisemmin kuin polkupyörällä. Tämäkin lautamalli on varustettu takarenkaan tehokkaalla levyjarrulla, jonka raju käyttö voi viskata ajajan jorpakkoon. Eturenkaan jarru ottaa talteen jarruenergiaa. Lisäturvaa tuovat takana kissansilmä ja jarruvalo, edessä kirkas led-valo.

Laitteessa itsessään on minimalistinen määrä ohjainlaitteita ja ainoa näyt-

tö kertoo akun latauksen. Lisää tietoja, toimintoja, laudan asetusten säätöä ja tuiki tärkeä lukitustoiminto löytyvät mobiilisovelluksesta.

Xiaomi on fiksu, kestävän oloinen ja hyvin suunniteltu sähköpotkulauta, joka jätti positiivisen vaikutelman. Se on varmasti kelpo vaihtoehto e-lautaa harkitseville, jotka haluavat laadukkaamman ja kestävämmän kulkuneuvon. Ja kun talvet ovat ainakin eteläisessä Suomessa jo varsin lyhyitä ja vähälumisia, niin kaksipyöräilykausikin vain pitenee koko ajan.

Ja jos Nintendo Switchillä pelaava lautailija kiitää jonnekin syrjempään, niin *Bionik Power Platella* konsolissa ja ohjaimissa piisaa virtaa pidemmänkin retken pelailuihin. Näpsä laite toimii sekä konsolin matka-akkuna että ohjainten latauskahvana, minkä lisäksi sillä lataa myös kännykän tai tabletin. Simppelissä akkupaketissa onkin paljon kelpoisaa toiminnallisuutta.

Kerro se kuvin ja videoin

Kesä innoittaa myös kaikenlaisiin jänniin aktiviteetteihin. Luvassa on lennokkien lennätystä, vauhdikasta sähkökiitämistä, ulkomikroautoilua, retkiä ja jumiutumisia seikkailupuistojen köysiradoille. Joskus tilanteet täytyy saada talteenkin.

Kesän kameratarjonta on vähintäänkin monipuolista. Aloitetaan sniikeimmästä. *Bear Grylls Waterproof Video Eyewear* on yksinkertaisesti aurinkolasit, johon on upotettu huomaamaton videokamera ja mikrofoni. Videokamera on siten käytännössä täysin näkymätön, onpahan vain kaverilla tyylikkäät aurinkolasit.

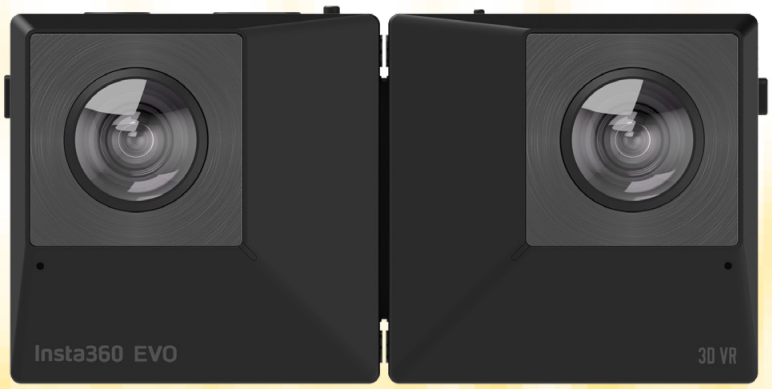
1080p30-kuvaa 90 asteen kuvakulmalla tallentava video- ja stillikamera ei ole ihmeellinen, mutta sentään laadukkaampi kuin keskimääräiset Kiina-romut. Äänenlaatukaan ei säväytä, mutta oman puheen ja lokin kirkunan erottaa. Vähän nopeammin edetessä tuulen suhina osuu ikävästi mikrofoniin.

Minulle suurin ongelma oli lasien sovittaminen päähän. Pääni on näemmä keskivertospeksejä suurempi, joten sangat kiristivät ja pusersivat lasit ikävästi alas nenänpäähän, jolloin aurinkokin helottaa suoraan silmään. Kiristettävä turvanauha kuitenkin pidätti lasit paremmin paikallaan, ja sopivasti kiristettynä tuntui, että laseja voi pitää päässä hurjemmassakin röykytyksessä.

Lasien mukana tulee kirkkaat vaihtolinssit, joten niitä voi käyttää myös varjossa ja sisävalaistuksessa, mutta mikään pimeäkamera tämä ei ole. Lasit kestävät jonkin verran kosteutta ja upottamista, ja sisäiselle 16 gigatavun muistille mahtuu noin 2,5 tuntia videota. Yhdellä latauksella kuvaa tunnin.

Astetta järeämpään kuvailuun sopii *GoPro 7 Black*. GoPro ei ole suotta synonyymi actionkameroille, sillä parhaimmillaan tämä on järisyttävän hyvä videokamera. Kuvanlaatu on upea, mikrofoni ensiluokkainen ja kuvaustiloja riittää 4K–720p-väliltä. 1080p200 on jo varsin näppärä kovaan actioniin. Parasta on HyperSmooth-kuvanvakautin: tärisevään ja heiluvaan kypärään, tankoon tai muuhun alustaan kiinnitettynä kuva pysyy tarkkana ja vakaana. En olisi uskonut, että näin hyvään kuvanlaatuun voisi päästä ilman ulkoista vakautinta.

Harmi vain, että pulmiakin riittää. Välillä kamera lopettaa tallentamisen itsestään. Akkukesto on surkea, joten ilman lisäakkuja ei pärjää. Kovempien kuvaustilojen formaatit eivät avaudu vanhemmissa videonkäsittelyohjelmissa. Toimintanappuloiden sijoittelussa ja fyysisessä designissa on tyhmiä ratkaisuja. Black lupasi myös livestriimauksen suoraan kamerasta, mutta sen toimivuus ja toteutus on surkea, lähes hyödytön. Kännykällä striimaa kätevämmin, vaikka luuria ei saakaan kiinni kypärään. GoPro 7 Black onkin monin osin paha pettymys. Jos et himoitse HyperSmoothia, niin älä vaivaudu.



Kesän vinkein kuvauslaite on *Insta360 EVO*, joka yhdistää samaan laitteeseen 360- ja 3D-kameran. Tämä on toteutettu saranoimalla kamera niin, että kameran voi avata tai sulkea kuvaustarpeiden mukaan.

360 asteen 5K-videointi 30 kuvaa sekunnissa -nopeudella oli kiitettävän tarkka ja selkeä. Kameran ohjelmistokuvanvakautus toimii upeasti esimerkiksi Thetan 360-kameraan verrattuna. 3D-kuvausta en ehtinyt kokeilla Skrollin aikataulujen puitteissa. Insta360:n ilmainen 360/3D-videoeditointiohjelmisto oli hyvä ja toimiva tarkoitukseensa.

Eniten EVOssa mietityttää saranakiinnityksen ja lukituksen pitävyys. Kamerassa on pelkästään tavallinen ruuvikierrekiinnitys, eikä ainakaan valmistaja tarjoa rytyytyskiinnikkeitä. Niinpä sen kiinnittäminen rajumpia 360-kuvauksia varten vaatii paljon gaffa- tai jeesusteippiä. Ei ehkä se elegantein ratkaisu. Muutoin niin hauska ja toimiva laite, että tänä kesänä kuvaan 360-videota ja paljon.

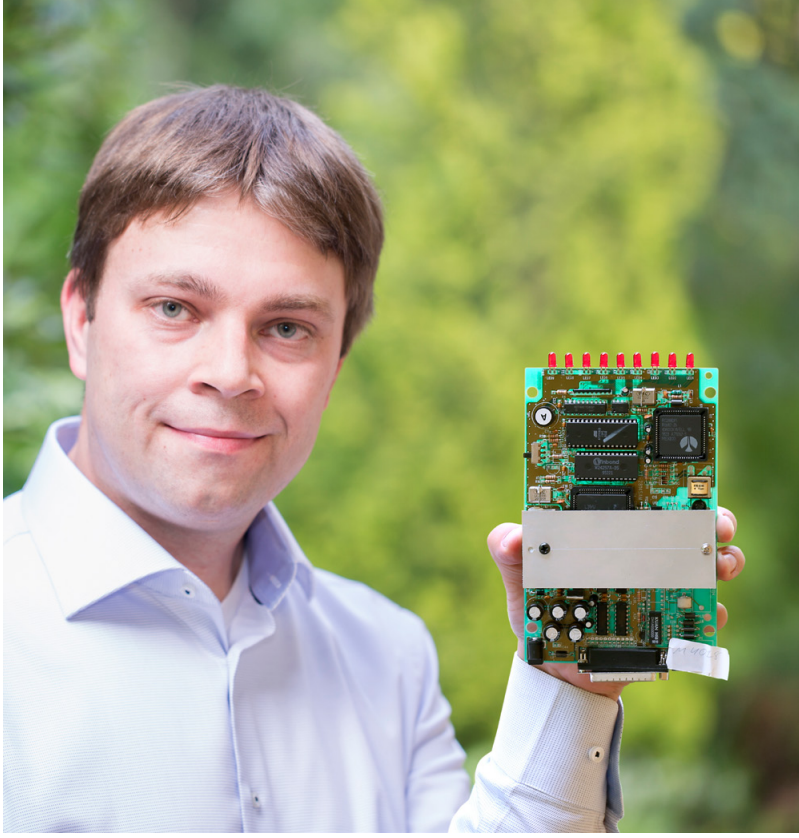
Esitellyt kapineet ovat kirjoittajan omia tai lainaamia testilaitteita.

Bear Grylls Waterproof Video Eyewear
200 €
Bionik Power Plate (Nintendo Switch)
40–50 €
GoPro 7 Black
300–400 €
Insta360 EVO
450–475 €
Quadcopter (Lidl)
27 €
Xiaomi Mi Electric Scooter M365
470–500 €

Epsilon Indi BBS

Sosiaalinen media on tehnyt verkkokeskustelusta epäsosiaalista.

Janne Sirén



Ylläpidin vuosina 1991–1996 Epsilon Indi BBS -nimistä purkkia, *bulletin board systemiä*. Aikana ennen internetin valtavirtaistumista, BBS:t olivat suosittuja teksti- ja merkkigrafiikkapohjaisia ”verkkosivuja”, tietokoneita lankapuhelinlinjan päässä. Käyttäjät ottivat purkkeihin yhteyttä omalla tietokoneellaan, hyödyntäen koneen ja puhelinlinjan väliin kytkettävää lisälaitetta: modeemia.

Purkeissa käyttäjät hakivat sekä jättivät viestejä, tiedostoja ja vuoropohjaisten pelien siirtoja. Purkin samanaikaisten käyttäjien määrä riippui sen puhelinlinjojen määrästä. Nykyvertailukohta voisi olla yksi SIM-kortti jokaiselle samanaikaiselle käyttäjälle. Tuolloin se vain tarkoitti lisäksi yhtä fyysistä kuparikaapelinippua, per käyttäjä, lähimpään puhelinjakamoon tai -keskukseen.

Suurimpiin purkkeihin mahtui kymmeniä tai satoja samanaikaisia käyttäjiä, mutta useimpiin vain yksi tai kaksi. Epsilon Indiinkin oli vain yksi puhelinlinja. Suunnitelmia toisesta linjasta oli, mutta ne eivät toteutuneet. Lisäksi ainokainen linja oli poissa käytöstä silloin, kun järjestelmän ylläpitäjä eli minä käytin linjaa tai tietokonetta johonkin muuhun, mikä sekin oli tuolloin tavanomaista.

Silti Epsilon Indiin – ja lukuisiin sen kaltaisiin purkkeihin vuodesta 1978 alkaen – syntyi sosiaalinen ekosysteemi, jollaisesta voi nykyisin vain unelmoida.

BBS:t ja uutisryhmät

On pienoinen ihme, että Epsilon Indiä käytti kukaan. Jälkiviisaasti sen tehtävä kun vaikuttaa olleen lähinnä oman kokeilunhaluni tyydyttäminen. BBS:t toimivat yleensä valmisohjelmistoilla, joita kutsuttiin purkkiohjelmistoiksi. Epsilon Indi kävi viiden ja puolen vuoden aikana lävitse lukuisia ohjelmistoja: *TransAmiga*, *Falcon*, *Excelsior!* ja *TechnoBBS* Amigalla sekä *SuperBBS* ja *BBBS* PC:llä – kokeilin kulisseissa monia muitakin. Aina kun vaihdoin ohjelmistoa, Epsilon Indin viestialueet tyhjenivät ja käyttäjät joutuivat rekisteröitymään uudelleen.

Silti muutamasta kutsutusta ja purkkilistojen kautta paikalle eksyneestä käyttäjästä kasvoi yhteisö, joka kerta toisensa jälkeen täytti viestialueet keskustelulla, vaihtoi tiedostoja ja pelasi pelejä, kuten *Legend of the Red Dragonia*. Keskustelu kävi sen verran vilkkaana, että innostuin kehittämään jopa oman etälukuohjelmani *Offline Orbitin*, jota käytettiin pian Etelä-Amerikkaa myöten. Etälukijoilla viestejä pystyi lukemaan ja kirjoittamaan ilman jatkuvaa modeemiyhteyttä.

Etenkin keskustelu purkin viestialueilla – messuilu, kuten sitä kutsuttiin (*messaging*, ei tule sotkea Messengeriin ja mesettämiseen) – poikkesi suuresti nykypäivän sosiaalisesta mediasta. Siinä missä nykysome koostuu lähinnä emojeista ja tekstiviestin mittaisista huudahduksista, tyypilliset viestit Epsilon Indissä ja muissa purkeissa muistuttivat kirjeitä. Keskustelu oli syvällistä tavalla, johon harvoin enää törmää.

Leimallista messuilulle olivat myös asia-aiheet. Epsilon Indin, kuten muidenkin purkkien viestialueet oli jaettu useisiin alialueisiin tekniikasta viihteeseen. Vaikka käyttäjät tulivat toisilleen tutuiksi, ja tietty ydinjoukko piti keskustelun vireänä, nykysomelle leimallista henkilöbrändäystä ei juuri esiintynyt – ei ainakaan demoskenen ulkopuolella. Puhuttiin asiaa asioista.

Lähin vertailukohta purkeille olivat internetin tai Usenetin UUCP- ja sittemmin NNTP-protokollaa käyttävät uutisryhmät (*newsgroups*), jotka avattiin vuonna 1980. Uutisryhmät ammensivat yliopisto- ja tiedemaailman perinteestä, jossa asiakeskustelua ja tietoa oltiin totuttu jakamaan kirjeissä ja ilmoitustauluilla (*bulletin board*). Kun reaaliaikaisuus puuttui, viestintä oli harkitumpaa.

Web-foorumit

Moderniin internet-aikaan purkki- ja uutisryhmäperinne jatkui web-sivujen foorumien muodossa 1990-luvun lopulla. Ei ole sattumaa, että foorumiohjelmistojen nimissä vilahtelevat sanat ”bulletin board”. Web-foorumit omaksuivat monia purkkien ja uutisryhmien ominaisuuksia, kuten viestiketjut ja visuaaliset sitaatit.

Kyky ketjuttaa viestejä ja siteerata niitä onkin edelleen olennainen osa verkkokeskustelua. Ketjutettu keskustelu mahdollistaa pitkät ajatuksenvaihdot, jotka tosin johtavat kirjailija **Mike Godwinin** ”lain” (*Godwin's Law*) mukaan aina lopulta Hitler- tai natsiviittauksiin – Godwin teki havaintonsa uutisryhmissä. Suomessa voisin ehdottaa talvisotakorttia.

2000-luvun alussa näyttikin siltä, että web-foorumi jatkaa sitä, mihin uutisryhmät ja purkit jäivät. Toisin kuitenkin kävi, kuten Skrollin toimittaja **Ronja Koistinen** kirjoitti jo Skrollin 2016.1 kolumnissaan. Ronja kertoi, kuinka erään hänen seuraamansa yhdistyksen kerran niin aktiivinen webfoorumi näivettyi ja suljettiin, kun käyttäjät karkasivat Facebookiin. Tarina on tuttu.

Mikään web-foorumi ei ole ilmiölle immuuni, mutta etenkin pientä tai uutta web-foorumia on ollut 2010-luvulla vaikea saada lentämään. Yksi syy

on tietoverkkojen käytön siirtyminen mobiiliin. Kun viestejä vaihdetaan yhä useammin älypuhelinsovelluksella, eniten huomiota saavat ne sovellukset, jotka keräävät laajimman käyttäjäkunnan yhteen sovellukseen. Monille näitä ovat Facebook, Twitter tai Instagram. Ei siinä enää kirjauduta erikseen jollekin foorumille.

Maailman muuttuminen olisi tietenkin ollut Epsilon Indinkin kohtalona, ellen olisi lopettanut sitä ajoissa keskittyessäni muihin harrastuksiin. Miksipä kukaan ottaisi yhteyttä modeemilla erilliseen purkkiin, kun ottamalla yhteyttä omaan internet-palveluntarjoajaan pääsee käsiksi palveluihin ympäri maailmaa. Nykyään yhteys internetiin on vieläpä aina auki ja mukana taskussa.

Sittemmin seurasin maailman suurimman purkin kuihtumista MikroBitti-lehden MBnetin ylläpidossa, aina siihen hetkeen, kun sammutimme MBnetin punaisella napilla vuonna 2002. Tai yritimme. Avustaja **Tarmo Toikkanen** nimittäin näpräsi punaista seis-painiketta sammutuksen jälkeen ja MBnet nousi hetkeksi kuolleista. Varmistaaksemme, ettei Tarmo tee enempää biblikaalisia tekoja, irrotimme MBnetin modeemit muistoksi. Kolumnin valokuvassa on MBnetin VIP-linja 39.

En minä silti purkkeja haikaile, harrastelijapurkit olivat pienen piirin juttuja ja suurten tietoverkkojen mahdollisuudet ovat omaa luokkaansa. Muutamat vanhat BBS:t kummittelevat vielä tänäkin päivänä telnet-yhteyden päässä internetissä (telnetbbsguide.com), mitä voi pitää kulttuuritekona, mutta käytännön merkitystä niillä ei ole – eikä tarvitsekaan olla.

Suurempana säälinä myös monet web-foorumit uhkaavat kuihtua olemattomiin, vaikka niiden vaalimiseen olisi paljon ajankohtaisempia syitä.

Sosiaalinen media

2000-luvun alkukymmenellä noussut sosiaalinen media ei nimittäin houkuttele syvälliseen asiakeskusteluun. Ensinnäkin keskustelusta on tullut alisteista jollekin avaukselle: se on esimerkiksi kuvan, henkilön tai blogikirjoituksen kommentointia. Koska avaukset tulevat usein yksittäiseltä taholta, kuten Facebook-sivun tai Instagram-tilin pitäjältä, keskustelu jää yksipuoliseksi.

Globaali sosiaalinen media tavoittaa myös kerralla niin ystävät kuin viholliset, asiayhteyden tuntevat ja tuntemattomat, ymmärtäjät ja väärinymmärtäjät. Purkeissa väki tunsi usein toisensa ja uutisryhmätkin rajautuvat jonkin alueen harrastuneisiin. Twitter sen sijaan tavoittaa kerralla suuren osan maailman väestöä, joka tulkitsee viestiä miten sattuu. Tämäkään ei edesauta harkittua keskustelua. Yksityisemmät Facebook-viestitkin ovat vaarassa vuotaa pääuutislähetykseen asti.

```
From: Anu Seilonen
To:   Veli-Matti Vuorensyrja
Subj: MUI: kannattaako regata?
Date: 13-Feb-96 05:28
Area: Amiga
BBS:  Epsilon Indi

> Siirsin itselleni MUI3.1 paketin ja olen kokeillut sitä muu-
> taman päivän, onko kellään kokemuksia regatusta versiosta?
> kannattaako rekisteröidä tämä ohjelma? kuinka rekisteröinti
> sujui? etc...

 MUI:in ihanuuteen sen enempää puuttumatta (kolmosversio on
 poistanut paljon sitä aikaisempia versioita vaivannutta
 tahmaa): Olen aikoinaan rekisteröinyt 2.3:n, joka tuli ihan
 OK perille. Rahat vain kuoressa ja ohjelma paluupostissa.

 Kolmosversio vaati sitten päivitysmaksun, sen verran ovat
 katsoneet ohjelman harpanneen. Lähetin maksun normaalisti
 seteleinä hyvin suojatussa kirjekuoressa Ruotsin regsiteen
 marras-joulukuussa, mutta mitään ei ole kuulunut kyselyistä
 huolimatta. Taisi tulla ensimmäinen "huti". Todella paljon
 olen ohjelmia rekisteröinyt samalla tavalla, ja tämä on
 ensimmäinen hassi. Perkule. :(

 Onko muilla kokemuksia tuosta Ruotsin regsitestä?

-+- Thoriel -+-

... Campbell's Law: Nature abhors a vacuous experimenter.
___ Offline Orbit v0.80a (010)
```

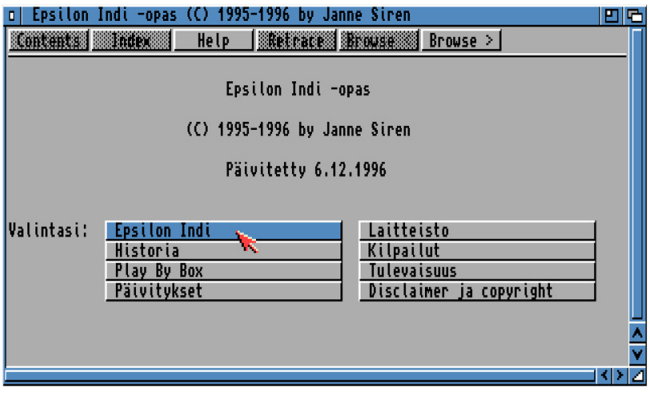
"Messuilua" Epsilon Indissä alkuvuodesta 1996.

Kolmas ongelma on, että viestinnän mittaa on usein rajoitettu, joko Twitterin tapaan merkkimäärillä tai tekemällä pidempi viestintä hankalaksi. Esimerkiksi Facebook lähettää kommentin nykyisin saman tien, jos yrittää lisätä tekstikappaletta pelkällä enter-näppäimellä, ja osa pitkistä viesteistä piilotetaan lukijoilta. Tämäkään ei houkuttele pitkälliseen viestintään, mikä lienee tarkoituskin.

Sielläkin missä voidaan viestiä pitkällisemmin, algoritmit pitävät huolen siitä, ettei keskustelu tavoita kaikkia tasapuolisesti, eikä toistuvaa ajatuksenvaihtoa juuri synny. Kaikki huutavat tyhjiöön – messuilulle niin olennaista jatkokeskustelua ensikommentin jälkeen syntyy harvoin. Jopa viestiketjuistaan tunnettu Reddit, jota voitaneen pitää web-foorumien perillisenä, rajoittaa viestiketjujen ajallista kestoa ja antaa äänestää viesteistä, jotka ylipäätään näkyvät lukijoille.

Klikkijulkisuus onkin sosiaalisen median pahin kirous. Koska sosiaalisen median algoritmien tehtäväksi on muodostunut aktiivisten silmäparien maksimointi mainostajille, esiin tuppaa nousemaan ennemmin heppoista viraalihittikamaa kuin syvällistä keskustelua. Algoritmit palkitsevat mekaanisista tykkäyksistä ja jaoista, eivät asiantuntevasta tiedosta tai perustellusta mielipiteestä. Puhumattakaan lukuisista lieveilmiöistä, joita tykkäystenkalasteluekosysteemi synnyttää.

Epsilon Indin ja MBnetin jälkeen olen ylläpitänyt muun muassa Skrollin sosiaalista mediaa viimeiset nelisen vuotta. Meillä on ollut eri aikoina käy-



Epsilon Indi -opas, jota ylläpidin eräänlaisena käyntikorttina purkista. Teknisesti opas on AmigaGuide-hypertekstiä. Skrollin verkkojatkoilla voit lukea oppaan myös HTML-muodossa.

tössä yhteensä kaksi Facebook-sivua, yksi Facebook-ryhmä, kolme Twittertiliä, kolme (ali)foorumia, Instagram-tili, YouTube-kanava, Reddit-alasivu, Snapchat ja pari IRC-kanavaa. Lähes kaikki käyttämämme sosiaalisen median alustat ovat suosineet helppojen teknologiavitsien ja halpojen retrofiilistelyjen julkaisemista – niillä saa kuulijoita.

Mikään käyttämistämme kanavista ei sen sijaan ole onnistunut tukemaan kestävää, syvällistä purkkihenkistä viestinvaihtoa. IRC reaaliaikaisena chattina on sentään säilyttänyt kriittisen massan läpi vuosien, arvatenkin kyseisen viestintäteknologian uskollisen kannattajakunnan ansiosta.

Loppusanat

Yksi asia kuitenkin yhdistää muinaisia purkkeja ja modernia somea: kokemusaineiston säilömisen vaikeus, jota **Vesa Linja-aho** päivitteli kolumnissaan Skrollissa 2016.2. Kuten muinaisessa Epsilon Indissä, jossa mielivaltaiset järjestelmämuutokseni ja satunnaiset haverit nollasivat purkin sisällön ajasta aikaan, samoin sosiaalisessa mediassa vanha sisältö tuppaa katoamaan tai vähintäänkin piiloutumaan jonnekin yksityisesti ylläpidettävien järjestelmien syövereihin, kunnes palvelu joku päivä lopetetaan.

Sen sijaan standardisoidut uutisryhmät ja selkeän URLin päässä olevat web-sivut ovat arkistoitavissa ja löydettävissä vielä vuosikymmenienkin päästä. Siinä missä Epsilon Indistä on jäljellä vain vähän lokia, Usenet-uutisryhmistä – jotka muuten toimivat edelleen – löytyvät yhä kaikki vanhat viestini ja ensimmäinen koskaan kirjoittamani artikkeli: kotikutoisen 14 megahertsin Amiga-kiihdyttimen testi comp.sys.amiga.reviews-uutisryhmään maaliskuulta 1992. Eri asia sitten on, tarvitsisiko sen enää löytyä.

Niin tai näin, menen tästä sometamaan videon, jossa kissa leikkii Commodore 64:llä – ja sinä menet painamaan tykkäysnappia.

Epsilon Indi -muistelmaopas, ensimmäinen artikkelini, Offline Orbit, edellä mainitut kolumnit sekä Skrollin sosiaalisen median linkit ja arkistot löytyvät verkkojatkoilta: skrolli.fi/numerot

Microsoft Surface Book 2

Jumit pois virtakaapelilla

Skrollissa 2018.4 **Ronja Koistinen** kertoi AMD Ryzen -suorittimien epävakaudesta Linuxissa, mikä ilmeni jumiutumisina käytössä ja yöksi päälle jätettäessä. Syyksi epäiltiin virransäästötiloja. Olen keväällä kärsinyt samoista oireista puolitoistavuotiaassa Surface Book 2 15” -kannettavassa. Siinä on kuitenkin Intel Core i7-8650U -suoritin ja Windows 10 Pro. Jumi on niin syvä, että resetointi vaatii virtanäppäimen pitkän painalluksen. Näytönohjainajurinkaan resetointinäppäimet eivät auta.

Ihmetykseni oli suuri, kun löysin jumitukseen ratkaisun: virtajohdon irrottamisen, jolloin kone siirtyy akuille. Riittää, että magneettista virtaliitintä vähän nostaa toisesta päästä irti. Jumituksia tapahtuu myös akkukäytössä, jolloin samaan pääsee kytkemällä töpselissä olevan virtakaapelin kiinni. Jumitus loppuu kuin taikaiskusta ja kone jatkaa siitä mihin jäi – tiedot eivät siis katoa!

Ainakin yksi käyttäjä on raportoinut samasta ratkaisusta Microsoftin foorumilla lokakuussa, mutta sai vastauksena ympäripyöreitä päivitys- ja puhdistusohjeita. Epäilykseni on, että jumitus liittyy lämmönhallintaan tai virransäästötiloihin, joita virtakaapelikin säätelee. Jos tiedät mistä on kyse, tai haluat kirjoittaa lehteen virransäästötilojen haasteista, ota yhteyttä: toimitus@skrolli.fi.

Janne Sirén

Päivitys: Mitä tapahtui kotirobotille?

Kotirobotin lobotomia

Kirjoitimme Skrollissa 2019.1 kotirobottien vaikeasta vuodesta 2018, jonka aikana yli puolen tusinaa nimekästä robottiprojektia kuopattiin. Esittelimme poikkeuksena lelurobottivalmistaja Ankin ja sen Anki Vector -robotin. Puolitoista kuukautta lehden julkaisun jälkeen kävi ilmi, että Ankikin oli ollut vain yhden epäonnistuneen rahoituskierroksen päässä lopettamisesta.

Kun Sony lopetti vanhojen Aibo-robottikoirien huoltamisen 2014, huolekseni jäivät lähinnä varaosat. Uusien pilvipalveluihin nojaavien robottien, kuten Anki Vectorin tai uusimman Sony Aibon (2018), kohdalla huolet voivat olla paljon suurempia: mitä tapahtuu pilvipalvelimille, joiden toiminnallisuuteen älyrobotit nojaavat?

BERGin Little Printer – robottia muistuttava älytulostin, jonka valmistus lopetettiin myös vuonna 2014 – antaa toivoa. Ankin tapaan yhtiö ei romahtanut suoraan konkurssiin, vaan se ajettiin alas hallitusti. BERGin johtaja **Matt Webb** piti palvelimet päällä vielä yhdeksän kuukautta ja tuki Tumblr-bloginsa kautta harrastelijahakkereita korvaavien ratkaisujen kehittämisessä. Harrastelijaratkaisut eivät pysty kaikkiin jippoihin, mutta Little Printer printtaa edelleen.

Myös Ankin lopettaessa yrityksen johtoporras jatkaa ratkaisujen hakemista. Kyseessä lienee ostajan etsintä, jolta ei kannata toivoa liikoja. Lopetusuutisen jälkeen some täyttyi hätähuudoista: uudet käyttäjät eivät saaneet Anki-robottejaan enää aktivoitua. Totta tosiaan, tunnuspalvelimet menivät alas nopeasti. Aikaisemmin käyttöönotettu Anki Vector toimii kyllä vaikka ilman internet-yhteyttä, mutta isolla miinuksella: ilman edes perustason äänikomentoja. Toistaiseksi oman robottimme palvelinyhteys ja äänikomennot ovat toimineet.

Kun Vector vaikenee, korvaajaksi voi ostaa syyskuussa ilmestyvän *Lego Star Wars Boost Droid Commander* -älyrobopaketin. Lego-R2-D2:n voi tuen loppuessa purkaa ja osista rakentaa oivan SER-roskiksen.

Janne Sirén

Neurosyntetisaattori cellF miksaili aivan omiaan

NEUROVERKKO PÖRISI

Tiedekeskus Heurekassa sattuu ja tapahtuu kaikkea jännää. Biologisen neuroverkon ryydittämän neurosyntetisaattorin konsertti oli silti kummallisuuden mittapuulla omassa luokassaan.

Teksti: Jukka O. Kauppinen Kuva: Valhe Kouneli

CellF. Elävä entiteetti, astiassa kasvatettu, ei silti ihminen tai edes ihmismäinen, mutta silti muusikko. Australialainen taiteilija **Guy Ben-Ary** on luonut kollegoineen musikaalisen instrumentin, jossa ei ole oikein mitään järkeä, mutta sitäkin enemmän teknologista viehätysvoimaa.

CellF-neurosyntetisaattorin ydin koostuu neurokantasoluista, jotka on kasvatettu Ben-Aryn käsivarresta otetusta solunäytteestä. Näiden muodostama neuroverkko on järjestelmän ydin, tavallaan aivot, mutta ilman älyä – ja ilman tekoälyä. Tämä on biologiaa.

Solut on kytketty analogiseen syntetisaattoriin monimutkaisella ja herkällä järjestelmällä, joka syöttää livemuusikoiden soittamaa musiikkia. Entiteetti kuuntelee/tuntee/kokee musiikin, joka aiheuttaa soluastiassa sähköisiä reaktioita. CellF-laitteisto huomaa ja poimii nämä reaktiot, jotka lähetetään syntetisaattoreihin ja siitä edelleen äänentoistoon, jolloin läsnäolijat kuulevat pienen soluastian tulkintaa sen kuulemasta musiikista.

Neurosyntetisaattori itsessään ei ole niinkään artisti tai luoja. Ehkä se (hän?) on enemmänkin luomisen tuskassaan seonnut miksaaja, joka väänteleе vipujaan omien ja ainutlaatuisten taiteellisten visioidensa mukaan eikä piittaa pätkääkään lopputuloksesta.

Lopputulos on herkullisen hämmentävä. Konserttisalissa voi ja pitääkin liikkua, sillä eri kaiuttimet toistavat neuroverkon eri osien reaktioiden kautta suodatettua musiikkia. Lopputulos on pörinällä, pärinällä ja surinalla höystettyä musiikkia, jonka keskellä kuljeskelu muistutti hetkittäin surrealistisesta lempeää noise-konserttia.

”Se on elävä olento, ei mikään kone”, Guy Ben-Ary kuvaili instrumenttiaan. ”Se on tavallaan ulkoinen osa minua, se osa joka voi olla se rokkitähti, jota minusta ei koskaan tullut.”

Siitä kertoo tavallaan järjestelmän nimikin: cellF. Self. Minä.

Pieni solukko asuu kummalliselta jättikaiuttimelta näyttävän laite- ja johtorykelmän uumenissa. Koska cellF on ennen kaikkea teknis-taiteellinen projekti, niin tyylilliset seikat ovat olleet nekin tärkeitä. Neuroverkkosolukon design-yksiö on tarkoin ilmastoitu ja säädelty pieni asunto, jossa entiteetti pysyy elossa kaikissa olosuhteissa. Asuntoon on rakennettu myös musiikkistudio, jota voidaan liikutella konsertista toiseen.

”Neuroverkko toimii ihmeellisen samaan tapaan kuin syntetisaattorit. Ne molemmat reagoivat mikroskooppisiin sähkövirran vaihteluihin, joten solukon reaktioita voidaan käyttää sellaisenaan syntetisaattoreita ohjaavana datana. Kaikki tapahtuu millisekunneissa, joten dataa voidaan käsitellä analogisesti reaaliajassa, ilman tietokoneita.”

CellF on lumoava rakennelma, jonka inhimillinen, elävä osuus tekee elämyksestä kovin erikoisen. Se ei ole tiedostava eikä älykäs, mutta se reagoi ja muuttuu. Neuroverkko syntyy joka konserttiin uudelleen, jolloin se ohjelmoidaan/opetetaan tehtäväänsä, ja tulos on aina erilainen. Soluastian olento voi oppia kuulemastaan ja jopa luoda uutta dataa.

Ben-Ary kertookin, että erään konsertin jälkeisenä päivänä laite pörisi ja soi aivan itsekseen, ilman ulkoisia ärsykkeitä. Neuroverkko oli aloittanut aivan omat soidinmenonsa.

Tämä yllätyksellisyys tekee cellF:stä jotenkin inhimillisen. Vaikka se on loppujen lopuksi pelkästään älytöntä solukkoa, joka oli otettu pakastimesta vain 16 päivää aikaisemmin, niin se oli silti jotain aivan muuta kuin vain biologinen satunnaismeteligeneraattori. Olihan se kuitenkin eräällä tapaa osa edessämme istuvaa ystävällistä herraa. Kieltämättä tunnelma jännittyi, kun konsertin jälkeisessä keskustelutilaisuudessa kysyttiin, mitä pikku solukolle seuraavaksi tapahtuu. Jatkaako se elämäänsä kuplien ja pöristen vai miten?

Ei jatka. Pikku miksaajan elämä on lyhyt, onneksi sentään kylläinen.

Ben-Ary ynnää, että taiteestahan tässä on lopulta kyse. Taiteesta, joka kohtaa tieteen ja jonka tekemiseen on tarvittu myös tutkijoiden ja tieteenharjoittajien panosta. Seitsenhenkinen työryhmä rakensi CellF:ää usean vuoden mittaisessa projektissa. Tästä kymmenen kuukautta kului pelkästään ajatustyöhön, kun tiimi mietti, kuinka neurosyntetisaattori pitäisi ylipäätään rakentaa.

CellF ja Ben-Ary kollegoineen saapuivat Suomeen Biotaiteen seuran kutsumana.

Lisätietoja:
youtu.be/6KYxS9X19D0
youtu.be/M3zvWXUMxXE
youtu.be/iz3zvuvBYks
guybenary.com/work/cellf

CellF konsertoi Heurekassa jammaillen paikallisten artistien kanssa.

Ääntä kohti
Musiikinteko-ohjelmat

Skrolli ratsasi musiikintekijän varustekaapin.

Teksti: Mikko Möykky, Ville Ranki, Jarno Niklas Alanko
Kuvat: Jakke, Mikko Möykky, Ville Ranki

Musiikintekoon käytettäviä ohjelmistoja kutsutaan lyhenteellä DAW (*Digital Audio Workstation*). Nykyiset ohjelmistot kattavat kaiken sen, mihin ennen tarvittiin kokonainen studio. Moderneissa DAWeissa tulee mukana lähes aina sisään rakennettuja instrumentteja ja efektejä. Näiden lisäksi voidaan hankkia erikseen kolmansien osapuolien tekemiä liitännäisiä (*plug-in*). Ohjelmiston valintaan vaikuttaa, onko sillä tarkoitus äänittää, tuottaa, sovittaa vai miksata/masteroida, sekä tietokoneen käyttöjärjestelmä.

Mitä sitten tarvitaan, että tietokoneella voidaan tehdä musiikkia? Oikeastaan siihen ei tarvita kuin tietokone ja kaiuttimet tai kuulokkeet. Nuotteja voi piirtää hiirellä tai näppäimistöltä. Kätevimmin nuottien syöttäminen onnistuu erillisellä midi-koskettimistolla.

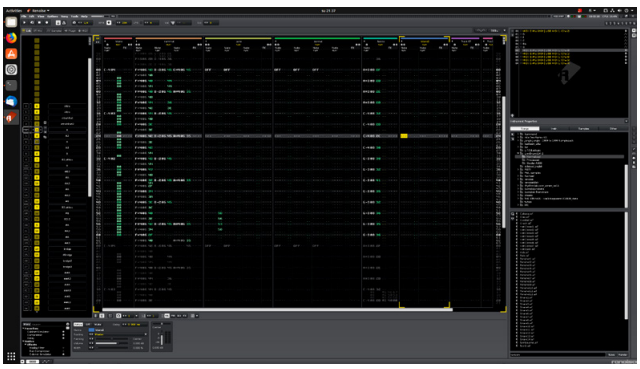
Melkein pakollinen hankinta kuitenkin on ulkoinen usb-äänikortti. Tämä pienentää DAWin latenssia huomattavasti. Jos kortissa on myös sisäänmenoliitäntöjä äänelle, sitä kutsutaan äänirajapinnaksi (*audio interface*). Kalliimmissa laitteissa voi olla monia sisään- ja ulostuloliitäntöjä, mikä mahdollistaa useamman äänilähteen samanaikaisen äänittämisen. Äänittäminen onnistuu myös normaalista tietokoneen mikkiliitännästä, mutta laatu on tyypillisesti huono.

Kun harrastus etenee, saattaa haluta hankkia ulkoisen rautasyntetisaattorin. Näitä voi yleensä ohjata midin kautta, mutta syntetisaattoria voi myös soittaa ja ohjata käsin sen koskettimistolta ja säätönupeista. Midi-data (ks. myös Skrolli 2018.3) kulkee audio interfacelle nykyään yleensä usb:n kautta, joskin vanhemmat mallit saattavat tarvita vanhan viisipiikkisen midi-liitännän, joka kuitenkin löytyy vielä useimmista ulkoisista äänikorteista. Syntetisaattorin tuottama ääni poimitaan tietokoneelle talteen audio interfacen kautta TRS- tai XLR-kytkennän kautta.

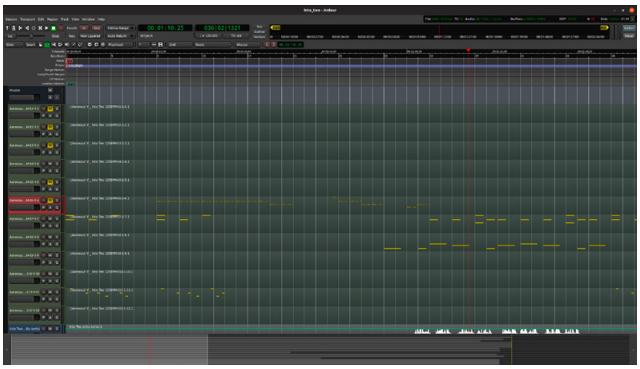
Yleiset ominaisuudet

Audion tallentaminen ja editointi: Jokaisen DAWin perusominaisuuksiin kuuluu äänen tallentaminen. Sitä voidaan äänittää joko yksi raita tai vaikka koko orkesteri kerralla. Ainoana rajoituksena on, kuinka paljon audio interfaceen saa kytkettyä soittimia tai mikrofoneja yhtä aikaa. Nykyiset audioraitohin perustuvat DAWit pystyvät käsittelemään satoja raitoja kerrallaan. Audioklippejä voidaan leikata, kopioida, loopata tai manipuloida niin paljon kuin mielikuvitusta riittää. Äänitteestä voidaan poistaa kohinoita tai sinne kuulumattomia räsähdyksiä ja poksahduksia.

Sekvensseri: Syntetisaattoreiden ja muiden midi-instrumenttien sävelet voidaan sekvensoida DAWiin miditietona. Näin soittoa voidaan jälkikäteen muokata yksi nuotti kerrallaan tai vaikka vaihtaa sävellajia. Myös rummut voidaan kirjoittaa midi-tietona vaikka yksi isku kerrallaan. Jokaiselle nuotille määritellään oma äänenvoimakkuus ja panorointi. Midi-tietoa voidaan syöttää joko perinteisesti midi-koskettimistoa tai muuta kontrolleria käyttäen tai yksi nuotti kerrallaan hiirellä syöttäen.



Renoise



Ardour

Mikseri: Jokaisesta DAWista löytyy virtuaalinen mikseri, jolla voidaan säätää jokaisen kanavan äänenvoimakkuutta ja taajuustasapainoa. Mikserikanaville voidaan lisätä efektejä, joilla manipuloidaan ääntä. Tyypillisiä efektejä ovat erilaiset dynamiikkaefektit, filtterit, kaiut, viiveet ja säröt. Efekteille voidaan tehdä automaatiota joko sekvensserin graafisella editorilla tai nauhoittamalla midi-ohjaimen liukuja tai nuppeja.

Virtuaali-instrumentit

Virtuaali-instrumentit ovat DAWeihin liitettäviä liitännäisiä tai itsenäisiä ohjelmia, jotka on ohjelmoitu mallintamaan syntetisaattoreita, signaaliprosessoreita tai midi-efektejä. Nykyisin lähes kaikista fyysisistä soittimista on virtuaali-instrumenttimallinnukset. Ohjelmoitujen synteesien lisäksi löytyy myös näytepohjaisia soittimia, jotka toistavat ennakkoon äänitettyjä äänipätkiä. Jokaista nuottia kohden saattaa olla äänitetty useita näytteitä monilla erilaisilla artikulaatioilla.

Näin pystytään soittamaan esimerkiksi orkesterisoittimia tietokoneella. Kokonaista jousisektiota on kohtuullisen helppo soittaa, mutta esimerkiksi yksittäisen viulun uskottava soittaminen vaatii hyvin huolellista askartelua artikulaatioiden ja yksityiskohtien kanssa. Taitavasti rakennettua raitaa voi olla vaikeaa erottaa aidosta äänitteestä. Yleisimpiä virtuaali-instrumenttiformaatteja ovat VST (*Virtual Studio Technology*), joka toimii lähes jokaisessa DAWissa, Applen oma formaatti AU (*Audio Units*) ja Avidin AAX-formaatti Pro Tools -ympäristöön.

Softa ja rauta

Erilliset äänikortit tarjoavat parempaa äänenlaatua ja enemmän ominaisuuksia. Musiikin teon kannalta on oleellista valita käyttöön sopiva kortti. Mikäli DAWia käytetään ulkoisten instrumenttien tai mikrofonien äänittämiseen, tarvitaan paljon sisäänmenokanavia sekä hyvät etuasteet. Jos työskennellään pelkästään ohjelmistolla, ei tarvita fyysisiä kanavia kuin kaiuttimia varten. Äänikortin laatu vaikuttaa myös äänen latenssiin eli aikaan, joka kuluu koskettimen painalluksesta siihen, että ääni toistuu kaiuttimista. Sekunnin kymmenyksenkin latenssi tekee midi-koskettimiston soittamisesta hankalaa. Varsinkin äänityskäytössä on erityisen tärkeää pieni latenssi.

Windows ja macOS musiikinteko ympäristöinä eivät eroa toisistaan paljoa. Ainoastaan jotkut DAWit ja virtuaali-instrumentit toimivat vain jommassakummassa. Molemmissa käyttöjärjestelmissä on ongelmana vanhojen laitteiden tukemisen lopettaminen. Uusia laitteita tulee niin vauhdilla valmistajilta, että vanhojen ajureiden päivitys loppuu aina jossain kohtaa. Windowsin etuna macOS:ään nähden on lukemattomat ilmaiset virtuaali-instrumentit. Linuxiin liittyy muutama jippo, joita käsittelemme oheisessa laatikossa.

Seuraavaksi esitellään tärkeimpiä musiikkiohjelmistoja, joihin kannattaa tutustua. Youtubesta löytyy hyviä tutoriaaleja eri ohjelmista, ja niitä kannattaa katsoa ennen käyttöä. Jokainen ohjelmisto on ”ammattityökalu” siinä mielessä, että ne vaativat opiskelua ennen käyttöä, varsinkin jos muiden vastaavien ohjelmistojen käyttö ei ole tuttua.

Renoise

Renoise on vuonna 2002 julkaistu DAW, joka muistuttaa tracker-tyylisiä musiikkiohjelmia. Se on trackerien kanssa yhteensopiva sen verran, että osaa esimerkiksi ladata erilaisia moduuliformaatteja 90-luvulta. Soittimina se kuitenkin osaa käyttää tavallisten samplejen lisäksi softasynia ja ohjata midi-laitteita. Kanavia on käytännössä rajoittamaton määrä ja niitä voi yhdistää ryhmiksi. Yksittäisille kanaville ja kanavaryhmille voi lisätä erilaisia efektejä, joita tulee Renoisen mukana iso kirjasto. Suosittelemme varauksetta Renoisea, jos tracker-tyylinen DAW tuntuu omalta tavalta tehdä musiikkia.

www.renoise.com, 68 € + verot / ilmainen demoversio
Windows/Linux/Mac

Ardour

Ardour on vuonna 2005 julkaistu moniraitanauhoituksiin keskittynyt DAW. Ardouriin kannattaa tutustua varsinkin, jos tarkoitus on tallentaa ja miksata perinteisillä soittimilla tuotettua musiikkia tai muuten tallentaa musiikki raita kerrallaan. Se tukee kuitenkin myös midi-raitojen tallennusta ja muokkausta. Ardour on ilmaisohjelmaksi varsin monipuolinen, ja esimerkiksi audiota pystyy reitittämään hyvin vapaasti tulojen ja lähtöjen välillä.

ardour.org, ilmainen
Windows/Linux/Mac

Audacity

Audacity on vuonna 2000 julkaistu ilmainen äänieditori ja nauhoitusohjelma. Se on varsin helppokäyttöinen, ja sitä voi käyttää esimerkiksi samplejen muokkaukseen sekä lopullisen biisin nauhoitukseen ja tallennukseen. Audacityssä on iso valikoima liitännäisiä, joilla ääntä voi muokata. Useimmiten käytetyt liitännäiset ovat esimerkiksi äänenvoimakkuuden normalisointi, häivytykset ja kompressointi.

www.audacityteam.org, ilmainen
Windows/Linux/Mac

Bitwig Studio

Bitwig Studio on vuonna 2014 julkaistu yleiskäyttöinen ja ammattilaisille suunnattu DAW. Demon perusteella Bitwig vaikuttaa erittäin laadukkaalta ohjelmistolta, jolla onnistuu ”kaikki”. Bitwigin mukana tulee kattava valikoima efektejä ja softasynia. Kannattaa tutustua, jos tekee musiikkia ammatikseen tai on valmis maksamaan ammattityökalun hinnan.

www.bitwig.com, 379 € / ilmainen demoversio

Windows/Linux/Mac

Rosegarden

Rosegarden on vuodesta 1993 asti kehitetty DAW. Se on perinteinen midisekvensseri ja nuotinnusohjelma. Rosegarden kaipaa ulkoisen äänilähteen kuten softa- tai rautasynan joka tuottaa varsinaisen äänen. Rosegarden tukee sekä midi- että audioraitoja, eli sitä voi käyttää myös moniraiturina. Ilmaisena ohjelmistona Rosegardenia on helppo kokeilla, mutta se ei ominaisuuksiltaan vastaa ammattilaissoftia. Sopii kuitenkin kevyeen sävellystyöhön.

www.rosegardenmusic.com, ilmainen

Linux

Logic Pro X

Logic oli alunperin Emagic-firman tuote, joka julkaistiin 1993, mutta vuonna 2002 Apple osti Emagicin. Kauppojen jälkeen Logic 5 jäi viimeiseksi PC:llä toimivaksi versioksi. Logic on yksi käytetyimmistä DAWeista tällä hetkellä. Applella on myös Logicin audio engineä käyttävä Garage Band, joka tulee käyttöjärjestelmän mukana. Logicia on kehuttu sen todella laadukkaista natiiveista instrumenteista ja efekteistä.

www.apple.com/logic-pro/, 230 €

Mac

Ableton Live

Ableton julkaistiin vuonna 2001 ja se oli alkuun tarkoitettu etupäässä livenä soittamiseen, mutta kehityksen myötä siitä on tullut täysiverinen DAW. Vasta version 4 jälkeen vuonna 2004 Abletonissa pystyi käyttämään kolmansien osapuolien plugineja. Ableton on tällä hetkellä yksi suosituimmista DAWeista.

www.ableton.com, 79€, 349 €, 599 € versiosta riippuen

Windows/Linux (Wine)/Mac

Reason

Reason on vuonna 2001 julkaistu DAW. Se eroaa muista DAWeista paljon, sillä siinä instrumentit ja efektit ovat kuin räkki studiolaitteita, joihin voi tehdä kytkentöjä johdoilla, kuten oikeissa fyysisissä laitteissa. Reason oli pitkään suljettu ympäristö, eikä siinä voinut käyttää muita kuin sen omia instrumentteja ja efektejä. Vasta vuonna 2017 version 9.5 myötä tuli tuki kolmansien osapuolten plugineihin.

www.propellerheads.com, 349 €

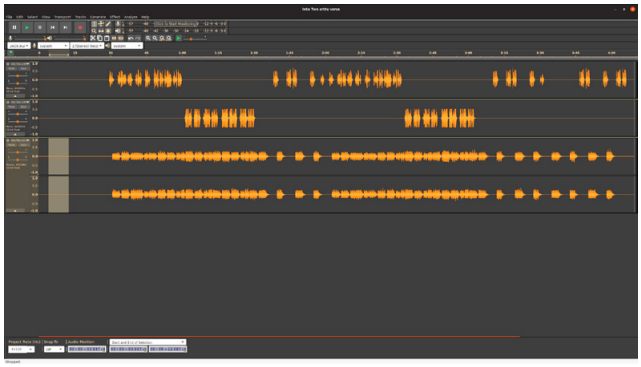
Windows/Mac

FL Studio

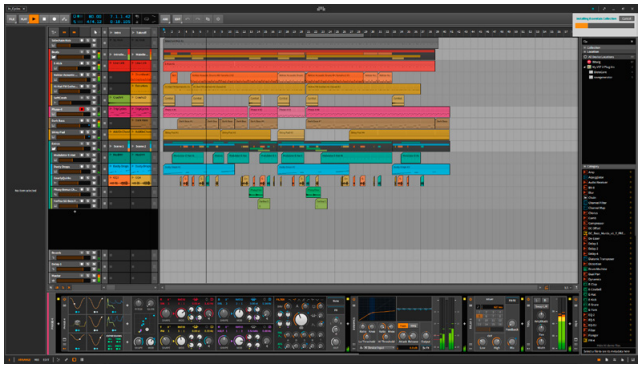
FL Studio, aikaisemmin Fruity Loopsina tunnettu DAW on julkaistu 1997. Ensimmäinen versio oli pelkästään rumpukone, mutta nykyisin se on täysiverinen DAW. Se on käytettävyydeltään helposti lähestyttävä verrattuna muihin DAWeihin. FL Studio toimii tällä hetkellä ainoastaan Windowsissa, mutta siitä on ollut pitkään jo kehitteillä macOS-versio, josta on beetaversio ollut testattavissa.

www.image-line.com, 89 €; 189 €; 289 €; 810,37 € versiosta riippuen

Windows/Linux (Wine)/Mac



Audacity



Bitwig Studio

Reason

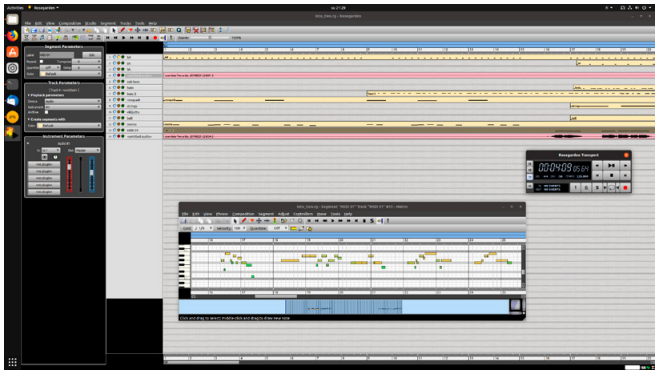
Logic Pro X

Pro Tools

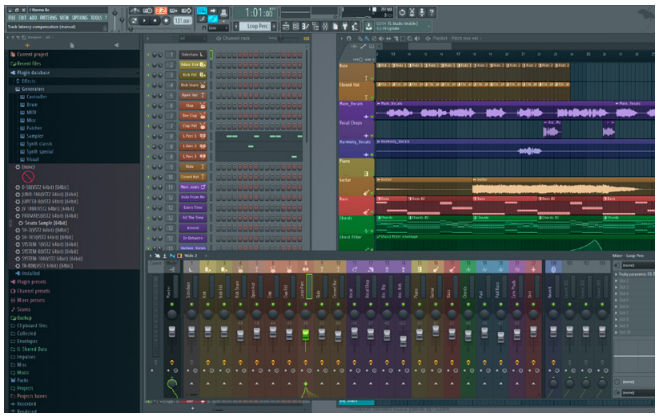
Jo vuonna 1989 julkaistu Avidin Pro Tools, eli tuttavallisesti ”Protari”, on yleisin DAW ammattimaisissa studioissa. Ohjelma tukee laajaa kirjoa erilaista rautaa ja on hyvä monimutkaisten studiojärjestelyjen toteuttamiseen. Ohjelman vahvuus on äänittämisessä, mutta midin kanssa työskentely onnistuu myös.

www.avid.com/pro-tools, 279 €/vuosi tai 559 € kertamaksu sisältäen päivitykset yhdeksi vuodeksi.

Windows/Mac



Rosegarden



FL Studio

Linux

Raudan kannalta Linux on sekä hyvä että huono asia. Laitteet useimmiten joko toimivat ongelmitta ilman säätöä tai eivät ollenkaan. Omat kokemukseni ovat olleet positiivisia, tähän asti kaikki audio interfacet ovat toimineet ilman mitään säätöä. Kannattaa kuitenkin etukäteen varmistaa laitteiden toimivuus netistä ennen hankintaa. Linuxin etuna on, että vanhemmatkin laitteet toimivat, jos niille on joskus tehty ajurit. Moni varsin laadukas vanhempi laite on yhä täysin käyttökelpoinen, mutta valmistajat eivät ole tehneet ajureita uusimmille Windows- ja macOS-versioille. Tällaisia laitteita saa pilkkahintaan netin myyntipalstoilta, ja Linux-käyttäjä voi saada toimivan ammattitason laitteen pikkurahalla.

Tavallisessa työpöytä-Linuxissa on käytössä Pulseaudio-niminen äänirajapinta, mutta siitä puuttuu tiettyjä ominaisuuksia, minkä vuoksi audiotyöskentelyyn on kehitetty oma Jack-niminen rajapintansa. Jackin etuna on matala latenssi, jota pystyy optimoimaan Jackin asetuksista oman koneen suorituskyvyn mukaan. Jackilla pystyy reitittämään audio- ja midi-virtoja ohjelmasta toiseen. Jack myös synkronoi eri ohjelmistojen mediakontrollit niin, että esimerkiksi moniraituri alkaa nauhoittamaan synkronoidusti, kun DAWista käynnistää soiton. Jackin graafinen käyttöliittymä on nimeltään QjackCtl.

Audiotyöskentelyssä kannattaa asentaa Linuxin ytimestä reaaliaikaversio. Sen avulla Jack ja sovellukset voivat pyytää reaaliaikaista moniajoprioriteettiä ja välttää häiriöitä äänentoistossa, jos kone kuormittuu liikaa. Ohjeet tähän löytyvät netistä.

Useimmat Linuxin audio-ohjelmistot tukevat ainakin VST-, DSSI- ja LADSPA-liitännäisiä. LADSPA (Linux Audio Developer’s Simple Plugin API) on rajapinta, jolla voi tehdä erilaisia ohjelmallisia efektejä. Linux-maailman softasynarajapinta taas on DSSI (Disposable Soft Synth Interface). Linux-jakeluiden pakettienhallinnoista löytyy helposti asennettava valikoima ilmaisia ja avoimia LADSPA- ja DSSI-liitännäisiä.

VST-maailmassa ”multiplatform” tarkoittaa harmillisen usein, että liitännäinen on tarjolla sekä Macille että Windowsille. Mutta tilanne ei ole toivoton, nimittäin nohevat hakkerit ovat keksineet keinot ”kääntää” Windows-liitännäinen Linux-liitännäiseksi. Käytännössä liitännäinen kääritään Linux-kirjastoksi, joka ajaa liitännäisen Winen avulla. Hyväksi on havaittu LinVst-niminen sovellus. LinVst:n käyttö vaatii jonkin verran komentorivin käyttöä ja ohjeiden lukemista, joten aivan aloittelijalle se voi olla hankala. Mainittakoon vielä, että osa Windowsissa toimivista DAWeista toimii Linuxissa Winellä.

Robottiauto. Nyt?

Sirénin sisäpiiri -palstalla Janne Sirén ruotii seuraamiaan teknologia-ilmiöitä ja -alakulttuureita. Tällä kertaa seuraamme itseajavia autoja.

Janne Sirén

Joulukuu, 2015: ”Arvioin, että meillä on täysautonomisia [autoja] noin kahdessa vuodessa.”

Tammikuu, 2016: ”Noin kahdessa vuodessa [Tesla-autojen etäkutsun] Summonin pitäisi toimia missä tahansa, missä on maayhteys, eikä rajoja. Esimerkiksi olet Los Angelesissa ja auto on New Yorkissa.”

Kesäkuu, 2016: ”Näkemykseni on, että olemme alle kahden vuoden päässä täysautonomiasta, ihmis[kuskeja] turvallisemmalla tasolla, mutta säännökset ottavat ainakin vuoden lisää.”

Huhtikuu, 2019: ”Jos ostat [itseajamiseen kykenevän] Teslan tänään, uskon, että ostat arvoaan kasvattavan esineen – ei arvoaan menettävän.”

– **Elon Musk**, Tesla[1]

Väittely sähköautoista

Sisäpiirin perinteeksi on alkanut muodostua aiheen raamittaminen mediakatsauksella, joten aloitetaan itseajavien autojen käsittely kysymyksellä: Pitäisikö tietokonelehden – saati tietokonekulttuurin erikoislehden – ylipäätänsä käsitellä autoja?

Tästä on ollut sekä Skrollin toimituksessa että lukijakunnassa monenlaisia mielipiteitä siitä asti, kun julkaisimme artikkelini sähköauton historiasta numerossa 2016.3. Artikkeli sai varsin polarisoitunutta lukijapalautetta. Toisten mielestä se oli numeron kiinnostavimpia juttuja, toisten mielestä se olisi kuulunut TM:ään. Ymmärrän molempia kantoja – juttu oli piilossa lehden sisäsivuilla tarkoituksella.

Kysymystä voi peilata Skrollin varhaisiin tehtävänkuvauksiin, joita löytyy **Ville-Matias Heikkilän** pääkirjoituksista. Skrollissa 2013.1 linjattiin, että tarkoituksemme ”on olla tietokonekulttuurin yhteinen lehti, jolle sekä kokonaiskuva että jatkuvuus ovat tärkeitä asioita. Se haluaa tuoda esiin kaikenlaisia tietoteknisiä tekemiskulttuureja jäämättä tiettyjen kuppikuntien äänitorveksi.”

Skrolli International Edition 2016.1E:n pääkirjoituksessa Ville-Matias linjasi – englanniksi – että ”keskitymme siihen, mikä tietotekniikassa on supistumatonta. Jos vain rennosti juttelet tai pelaat Go-peliä toisen henkilön kanssa internetissä, se ei varsinaisesti ole tietokonekulttuuria, koska voit tehdä sitä ilman tietokonettakin. Sama tekoälyllä, sen sijaan, olisi lähempänä ydintämme.”

Perinteisesti autot eivät ole olleet tietokonekulttuuria ja vielä harvemmin niitä näkee tietoteknisessä tekemiskulttuurissa – saati siellä, missä tietotekniikka on supistumatonta. Näin siitäkin huolimatta, että Skrollin tunnuslause ”Tietokonekulttuurin erikoislehti” on automaailmasta. Joku saattaa kuvitella sen olevan sanaleikki Mikrobitin ”Kotimikron (myöhemmin Tietokoneen) käyttäjän peruslehdestä”, mutta se on saanut inspiraationsa – luvalla – Mobilisti-autolehdeltä, joka on ajoneuvokulttuurin erikoislehti.

Ajoneuvomaailmassa on paljon arvokasta tekemiskulttuuria, mutta toistaiseksi se on harvemmin kohdannut tietokonekulttuuria.

[1] *Suomennos. Alkuperäiset sitaattilähteet sekä linkit artikkelin viittauksiin verkkojatkoilla: skrolli.fi/numerot*

Kuvamanipulaatio

Toinen vuosi

Tesla Autopilotin kyydissä

Skrolli kertoi viime kesänä (Skrolli 2018.2), miten ensimmäinen vuosi Teslan toisen sukupolven automaattiohjauksen matkassa sujui. Muuttuiko mikään vuonna kaksi?

Teslan julkaistessa toisen sukupolven nopeus- ja ohjausavustimensa Autopilotin loppuvuodesta 2016 se kohautti väittämällä järjestelmän raudan kyke-

Yhteiset supertähdet

Sähkö- ja itseajavien autojen myötä jotain olennaista vaikuttaa kuitenkin muuttuneen. Autot ovat olleet pyörillä liikkuvia tietokoneita jo vuosikymmeniä, mutta nyt niiden painopiste on siirtymässä esiteollisen polttomoottorimöhkäleen ympärille rakennetusta vaunusta muuhun – ja tämä *muu* on aikaisempaa lähempänä tietokonekulttuuria.

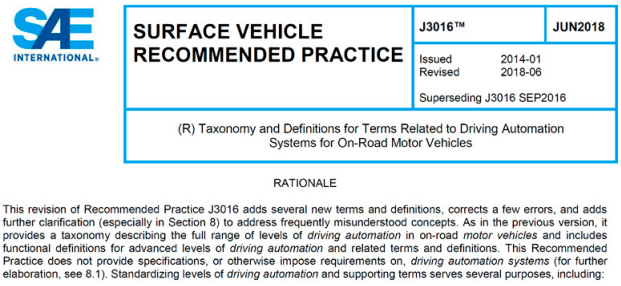
Autoissa elektroniikka ja yhä enenevässä määrin myös softa ovat nousseet pääosaan. Muutos rinnastuu siihen, kuinka puhelimesta on tullut tietokone. **Alexander Graham Bellin** ja kumppaneiden keksinnöllä ei ollut paljoakaan tekemistä tietotekniikan kanssa, mutta iPhone on jo asia ihan erikseen. Lähes kaikki tietokonemediat kirjoittavatkin nykyisin puhelimista – ja autoista.

Sähköautot ja itseajavat autot olivat esillä isosti myös Consumer Electronics Show'ssa Las Vegasissa tammikuussa. Venäläisen robottifirma Promobotin väitettiin jopa valjastaneen ne julkisuustemppunsa osaksi, kun yritys kertoi Tesla Autopilotin ja heidän robottinsa kolarista tapahtumapaikan lähellä. Autosta on tullut kuluttajaelektroniikkaa.

Tämä ei tosin ole kovin yllättävää toisestakaan syytä. Ihmiset seuraavat ihmisiä. Tietokonemediat kirjoittivat aikanaan Pixarista enemmän kuin *Toy Story* olisi ansainnut, koska **Steve Jobs** oli yhtiössä. Vanhoja Spectrum-lehtiä lukiessa on vaikea vältellä **Clive Sinclairin** Sinclair C5 -sähköauton (1985) mainintoja – *Your Sinclair* kutsui sitä kuuluisasti sähköiseksi bideeksi. Lehti vitsaili autosta vuosia.

Useat tietokonemaailman supertähdet ovat mukana auto- ja avaruushankkeissa. Kun SpaceX linkosi Tesla Roadster -auton Marsin suuntaan, se oli etusivukamaa tietokonemedioissa. Harva olisi kirjoittanut yhtä innokkaasti, jos NASA olisi lähettänyt Lotus Elisen avaruuteen. Autot ovatkin hivuttautuneet tietokonekulttuurin laitamille myös yhteisten viiteryhmien ansiosta.

Mikrobitti-lehdessä oli aikanaan säännöllinen *Peliluola*-palsta, joka käsitteli roolipelejä. Siis niitä paperisia, jotka liittyivät tietokoneisiin vain sen yhteisen viiteryhmän verran. Palstan kirjoittaja **Nordic the Incurable** kirjoitti samassa hengessä mainion artikkeliparin myös Skrolliin (2015.2–2015.3). Tätä menoa jonkun tulevaisuuden tietokonejulkaisun Peliluola onkin *Parkkiluola*.



SAE INTERNATIONAL

SURFACE VEHICLE RECOMMENDED PRACTICE

J3016™ JUN2018

Issued 2014-01
Revised 2018-06

Superseding J3016 SEP2016

(R) Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles

RATIONALE

This revision of Recommended Practice J3016 adds several new terms and definitions, corrects a few errors, and adds further clarification (especially in Section 8) to address frequently misunderstood concepts. As in the previous version, it provides a taxonomy describing the full range of levels of *driving automation* in on-road *motor vehicles* and includes functional definitions for advanced levels of *driving automation* and related terms and definitions. This Recommended Practice does not provide specifications, or otherwise impose requirements on, *driving automation systems* (for further elaboration, see 8.1). Standardizing levels of *driving automation* and supporting terms serves several purposes, including:

Robottiautoilun pyhä kirja. Yhdysvaltalainen standardisointijärjestö Society of Automotive Engineers on määritellyt itseajamisen tasot – Level 0-5 – dokumentissaan J3016. Linkki verkkojatkoilla.

Musk vastaan maailma

Edellisen Skrollin tekoälyteemaa työstäessämme huomasimme, kuinka vaikeaa aiheesta on kirjoittaa mainitsematta Muskia tai Teslaa. Ei siksi, että pitäisimme heitä varmoina tietolähteinä, mutta he ovat äänekkäitä. Heitä on vaikea sivuuttaa. He ovat kuin se kuvaannollinen junaonnettomuus – poispäin ei pysty katsomaan. Jälkeenpäinkin on pakko miettiä, oliko juna Autopilotilla.

Skrollin 2018.2 artikkelissa *Väittely robottiautoista kiihtyy* hahmottelin alan kaksi leiriä: Toisella laidalla ovat monisensoriteknologiaan (360 asteen Lidar-pulssilaserit, radiotutkat, kamerat, tarkkuuspaikannus) ja sääntöpohjaiseen kehitykseen luottavat vanhemmat robottiautoprojektit ja perinteiset autonvalmistajat. Vastassa ovat Musk ja uusyritykset, jotka vannovat perunakameroiden (edulliset 360 asteen kamerat ja vähän, jos lainkaan, muita sensoreita) ja koneoppimisen nimeen.

Maailma ei toki käytännössä ole näin mustavalkoinen: kaikki käyttävät konenäköä sekä koneoppimisen ja sääntöpohjaisten algoritmien yhdistelmiä. Tesla erottautuu kuitenkin muista väittämällä, että robottiauto tarvitsee nimenomaan *vain* kourallisen kameroita – ei kattavaa tutka- ja laserpeitettä. Teslallakin on yksinäinen tutka edessä

nevän tason 5 täysin itsenäiseen ajamiseen ohjelmiston laajamittaisen tarkistuksen ja viranomaisten hyväksynnän jälkeen. Teslan ja MobilEyen yhteistyö päättyi ja Tesla keskittyi kahdeksan kameran Tesla Vision -ratkaisuunsa.

Yhtiö aloitti samalla myymään *Full Self-Driving* -ominaisuutta (FSD) uusiin autoihinsa 8 800 euron hintaan (plus autovero), vaikka ominaisuus ei tehnyt mitään.

Suuret puheet

Sittemmin osaa Teslan alkuperäisiä Autopilot 2 -väitteistä lienee syystäkin kutsuttu vaporware-nimityksellä, ja koko Tesla Vision -projekti tiimeineen vaikuttaa menneen uusiksi parikin kertaa. Selvää on, ettei "tason 5 itsenäiseen ajamiseen" pystyvää järjestelmää ollut "tarkistettavaksi" asti vuonna 2016. Wall Street Journal -lehden lähteet Teslan sisältä väittävät, etteivät myöskään kaikki Autopilot-insinöörit arvostaneet toimitusjohtaja Elon Muskin suuria julkistuspuheita.

Järjestelmän toinen osa – *Enhanced Autopilot* -ohjausavustin (EAP, hinta tuolloin erillisenä 5 400 €) – sentään julkaistiin raakileena, mutta lennokkaasta tuotekuvauksestaan poiketen sekään ei päässyt edes Teslan ja MobilEyen vanhemman, puolet halvemman ensimmäisen sukupolven Autopilotin tasolle. Vuosikatsauksemme Skrollissa 2018.2 päättyikin siihen, että EAP-ohjausavustimesta oli tullut taipuvaisempi ohjaamaan päin liikenteenjakajia, eikä Autopilot käyttänyt kuin kolmea kameroistaan.

Lisäksi yhtiön vuoden 2017 loppuun lupailemaa itsenäisen ajon demoa Yhdysvaltain rannikolta toiselle lykättiin (ja huhujen mukaan myös yritettiin) toistuvasti. Myöskään *Tesla Networkista* – itsenäisesti ajavien Teslojen vuokrausverkostosta – ei

SAE-TASOT

Taso 0 (Level 0)
Ei ajoautomaatiota, korkeintaan varoituksia ja kuljettajan toimia tukevia ominaisuuksia.
Taso 1 (Level 1)
Auto säätelee yhtä ajamisen osa-aluetta, mutta ei useampaa samaan aikaan (esim. mukautuva vakionopeudensäädin nopeutta tai vanha pysäköintiavustin ohjausta). Kuljettaja vastuussa.
Taso 2, "Kädet pois"
Auto säätelee nopeutta ja ohjausta samanaikaisesti, mutta kuljettajan on oltava aina valmiudessa ja käsienkin yleensä kevyesti ratilla (esim. Tesla Autopilot) tai katseen tiessä kameran valvomana (GM Super Cruise, Nissan ProPilot 2.0).
Taso 2+, "Rahat pois"
Okei, keksin kuvauksen ihan itse, mutta keksitty on koko tasokin. Ei virallinen SAE-taso, vaan sekä Nvidian että MobilEyen alkuvuodesta CES:ssä esittelemä markkinointimääritelmä kehittyneille tason 2 ajoavustimille. Kuljettaja vastuussa.
Taso 3, "Silmät pois"
Rajatuissa tilanteissa auto on kokonaan vastuussa kohtuullisella varoitusajalla ja kuljettaja voi esimerkiksi lukea Skrollia (Audi Staupilot).
Taso 4, "Mieli pois"
Auto ajaa itsenäisesti sallituissa olosuhteissa, jopa ilman kuljettajaa kyydissä (esim. Waymo). Rattikin voidaan jättää pois rajatuissa tapauksissa, esimerkiksi kun autossa on etäohjaus (Sensible4).
Taso 5, "Ratti pois"
Auto ajaa itsenäisesti kaikissa olosuhteissa, joten ratti voidaan haluttaessa jättää kokonaan pois.

ja lyhyen kantaman ultraäänisensoreita, mutta Muskin mukaan tähtäin on pelkässä konenäössä.

Tesla on myös tuonut tuotteen ennakkomyyntiin ohjelmistopäivitettävissä autoissaan. Huhtikussa 2019, *Autonomy Investor Day* -sijoittajatapahtumassa, Musk löi lisää löylyä. Hän esitti, että Lidar-teknologiaan ja tarkkuuskarttoihin luottavat kilpailijat ovat tuhoon tuomittuja ja että Teslan täysin itsenäiseen ajamiseen kykenevä SAE-tason 5 (ks. taulukko) ohjelmisto, ilman maantieteellisiä rajoja, on ominaisuuksiltaan valmis jo kuluvan vuoden lopussa.

Vanha maailma

On vaikea sanoa, miten Muskin julistuksiin pitäisi suhtautua. Sisäpiirin alalaatikosta voitte lukea, millä tasolla Tesla on tänä päivänä. SAE-taso 5 tarkoittaa korkeimman autonomiatason autoa, jossa tietokone kykenee olemaan vastuussa ajamisesta kaikissa olosuhteissa, eikä kuljettajaa tarvita. Tällä hetkellä kaikki kuluttajille myytävät autot – myös Teslat – ovat korkeintaan tasolla 2, jossa auto hallitsee kahta parametria (nopeus ja ohjaus), mutta kuljettaja on aina vastuussa.

Toistaiseksi ainoa autonvalmistaja Teslan lisäksi, joka on kertonut olevansa lähellä kuluttajille suunnattua korkeamman tason järjestelmää, on Audi. Audi esitteli vuonna 2017 *Staupilot*-järjestelmän (*Traffic-jam Pilot*) uuden sukupolven Audi A8:n yhteydessä. Kyseessä on tason 3 järjestelmä eli auto on vastuussa ajamisesta rajatuissa tilanteissa ja luovuttaa vastuun takaisin kuljettajalle riittävällä ennakkovaroituksella. Kuljettaja voi esimerkiksi lukea kirjaa.

Staupilot toimii näin: Kun auto saapuu järjestelmälle sopivalle toiminta-alueelle (moottoritieruuhka, nopeus korkeintaan 60 km/h), kuvake ilmestyy mittaristoon. Kuljettaja kytkee järjestelmän päälle Audi AI -napista, jolloin valot ja äänet kertovat sen aktivoituneen. Auto on nyt vastuussa ajamisesta. Sen viihdejärjestelmä vapauttaa internetin ja televisionkatselun käyttöön.

SAE-tason 3 vaatimusten mukaan Staupilot hallitsee kaikki ajotilanteet toiminta-alueellaan, paitsi kuljettajalle ilmeiset auton vikaantumiset, kuten korin tai jousituksen yhtäkkisen rikkoutumisen. Pilotti osaa siis huomioida esimerkiksi moottoritietä ylittävät ihmiset ja eläimet sekä kaistaviivojen katoamisen. Se taitaa jopa *rettungsgassen*, saksalaisen tavan antaa tietä hälytysajoneuvoille.

Staupilot valvoo myös kuljettajaa, ettei tämä nukahda. Nukahduksen sattuessa tai toiminta-alueelta poistuttaessa (nopeuksien noustessa tai moottoritien päättyessä) Staupilot pyytää kuljettajaa rattiin noin kymmenen sekunnin varoajalla. Ensimmäinen heräte on kevyt ääni ja valo. Jos kuljettaja ei reagoi, tätä seuraa voimistuvia äänimerkkejä ja lopulta auto nykii turvavyötäkin.

Jos kuljettaja ei vieläkään reagoi, auto on edelleen vastuussa ajosta, mutta olettaa kuljettajan saaneen sairauskohtauksen ja ryhtyy pysäyttämään itseään. Hätävilkut kytkeytyvät, auto hiljentää vauhtia ja lopulta pysähtyy kokonaan kaistalle. Sisävalot syttyvät, ovien lukot avautuvat ja auto soittaa automaattisesti apua.

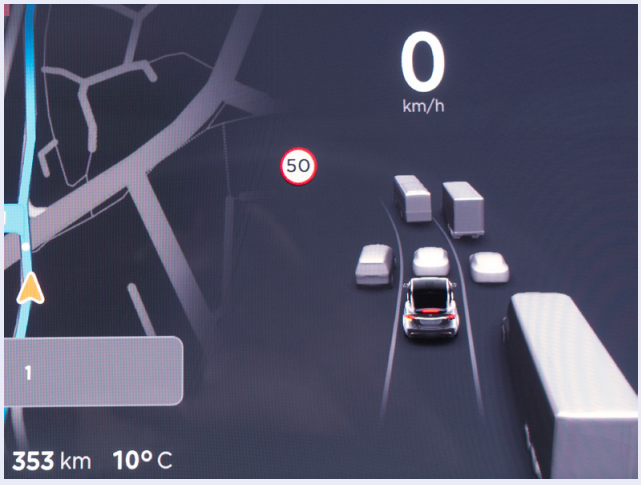
SAE-taso 3, toisin kuin tasot 4 ja 5, ei edellytä niin sanotun vähimmäisriskitilanteen saavuttamista itsenäisen ajon päätteeksi. Niinpä auto voi pysähtyä kaistalle, eikä sen tarvitse hakeutua esimerkiksi pientareelle. Staupilotia on esitelty toimittajille useasti ja se selvästikin toimii jo. Järjestelmä jumittaa hiiskuttu sanaakaan, vaikka osana Autopilotin tuotekuvausta lisätietoja luvattiin jo vuodelle 2017.

Hiljaisuus jatkui läpi vuoden 2018. Tässä vaiheessa homma alkoikin muistuttamaan tyypillistä joukkorahoituskampanjaa. Tesla palasi Tesla Networkiin vasta 22.4.2019 *Autonomy Investor Day* -tapahtumassa, siitä lisää pääjutussa. Rannikkoajokin sai uuden rohkean ennusteen: vuoden 2019 loppuun mennessä siihen pystyisivät asiakkaidenkin FSD-Teslat.

Mutta palataan vuoden 2018 alkupuoliskolle…

Ennen oli Pilotti, nyt pelotti

Toinen vuosi Tesla Autopilotin kyydissä alkoi ristiriitaisissa tunnelmissa. Tesla korotti FSD-paketin hintaa tonnilla, kunnes lopulta lokakuussa – kahden vuoden myynnin jälkeen – yhtiö lakkasi tarjoamasta sitä uusiin



Tesla näkee liikennettä Nihtisillalla, Espoossa.

kuitenkin vieläkin saksalaisessa säännösputkessa eikä sitä myydä autobaanoille, saati muualle.

Audilla on kehityksessä lisäksi *Highway Pilot*, joka toimii nopeuteen 130 km/h asti. Myös Nissanin *ProPilot 2.0* -järjestelmästä piti tulla tason 3 moottoritieavustin, ainakin Japanissa, mutta se julkaistiin toukokuussa Teslan *Navigate on Autopilot* -toiminnon kaltaisena tason 2 ratkaisuna.

Uusi maailma

Pisimmällä itseajavien autojen kehityksessä vaikuttaa olevan Googlen tytäryhtiö Waymo. Amerikassa itseajava auto onkin jo yli vuosikymmenen ajan tullut tunnetuksi katolla pyörivästä Lidar-kupolista. Myös esimerkiksi Uber – jonka testiautosta tuli viime vuonna ensimmäinen jalankulkijan tappanut tason 4 prototyyppiajoneuvo – ja General Motorsin Cruise-osasto (ei tule sotkea GM:n tason 2 moottoritieavustimeen Super Cruiseen) ovat tulleet tutuksi samankaltaisista robottiautoistaan.

Waymo tähtää tässä vaiheessa SAE-tasolle 4 eli itsenäiseen ajamiseen rajatuissa olosuhteissa. Sen Chrysler Pacifica -autoihin ja Jaguar I-Pace -sähköautoihin asennetussa sensorikokonaisuudessa on peräti 5 Lidaria, 4 tutkaa ja 9 kameramoduulia (noin 18 kameraa). Joulukuussa 2018 se käynnisti viimein robottitaksipalvelun, Waymo Onen, Phoenixissa. Toukokuussa 2019 kymmenen Waymon autoa liittyi Lyft-kuljetusverkostoon.

Yhtiön robottitaksit on toistaiseksi rajattu tarkkuuskartoitetulle maantieteelliselle alueelle – rajoitus, josta Musk teki pilaa puheissaan. Waymo tosin testaa monessa muussakin paikassa, joten merkittävä osa rajoituksesta on enemmän varovaisuutta kuin tekniikkaa. Google jos kuka osaa kartoittaa maailman. Waymoissa on normaalisti mukana myös turvakuljettaja, mutta ajoja on tehty myös ilman kuljettajaa. Turvakuljettajaa tarvitaan lähinnä jumi- ja pulmatilanteiden ratkaisuun.

Tason 4 itseajava sähköauto: Jaguar I-Pace, jossa Waymon robottiajo. Huomaa kiistelty Lidar-kupoli katolla. (Kuva: elisfkc/WmC)

Waymon haasteet eivät olekaan enää pitkään aikaan olleet aistimisessa, joka tietojen mukaan lähentelee täydellistä 3D-mallia ympäröivästä maailmasta. Ratkottavat aistimiskysymykset ovat enää hienovaraisia, kuten jalankulkijoiden liikkeiden ennustamista kehonkielestä. Isoimmat haasteet ovat ajopäätösten tekemisessä: Waymo on säätänyt autonsa hyvin varovaiseksi esimerkiksi epätavallisen ahtaissa manöövereissä tai käännyttäessä vasemmalle risteävän liikenteen editse.

Waymo tarjoaakin kutkuttavan ikkunan inhimilliseen käytökseen robottiauton äärellä. Pilottiasiakas **Diego Vera** kertoi olleensa kerran kyydissä ilman turvakuljettajaa – sillä matkalla hän ei uppoutunut kännykkäänsä, vaan seurasi auton tekemisiä herkeämättä. Ulkopuolella robottiautot ovat tosin herättäneet uteliaisuuden lisäksi tieraivoa. Autojen pilkuntarkka sääntöjen noudattaminen ja huolelliset manööverit eivät ole kaikkien mieleen.

Waymon turvakuljettajia on jopa uhkailtu ja autoja kiusattu jarruttelulla. Seassa vaikuttaa olevan jopa ripaus luddismia eli huolta siitä, että teknologia korvaa ihmiset. Myös Tesla kohtasi hieman vastaavan tapauksen huhtikuussa, kun erään **Randeep Hothin** väitettiin pörränneen sen testiauton ympärillä lähestymiskiellon veroisesti.

Vaikka Waymolla on eniten autonomisia kilometrejä taskussaan – sen juuret ulottuvat vuoden *2005 DARPA Grand Challenge* -kilpailuun – robottitakseja kehitetään muuallakin. Kiinassa ainakin Didi Chuxing on testannut autonomisia takseja ja Tokiossa ZMP:n ja Hinomaru Kotsun pilottirobotaksi ehti markkinoil-

autoihin, koska se ”aiheutti liikaa hämmennystä.” Vain EAP jatkoi julkisessa myynnissä, joskin FSD:n saattoi edelleen tilata tiskin alta.

Teslan myyjät varoittelivat asiakkaita, että FSD:n ilmestymiseen voi mennä ”todella pitkä aika.” Pian FSD:n ostaneet saivatkin seuraavan märän rätin kasvoilleen. Alkujaan Tesla oli ilmoittanut, että kahdeksan aktiivista kameraa vaativat FSD-paketin, muuten käytössä olisi neljä kameraa. Kahdeksan kameraa aktivoitiin kuitenkin kaikille ilman eri hintaa. Rätit eivät valitettavasti jäisi tähän.

Toisin kuin FSD, avustava automaattiohjaus EAP oli sentään Tesla-kuskien käytettävissä. Yksi keskeisistä moitteista oli kuitenkin, ettei EAP kyennyt samaan kuin kiitelty ensimmäisen sukupolven Autopilot, joka hyödynsi MobilEyen EyeQ3-piiriä Teslan omien neuroverkkojen sijaan. Ohjaus ei ole ollut yhtä luotettavaa ja nopeusrajoitusliikennemerkkien tunnistaminen sekä vilkusta käynnistettävät automatisoidut kaistanvaihdot moottoriteiden ulkopuolella puuttuivat EAP:sta.

Onneksi Teslat saava ohjelmistopäivityksiä. EAP:n päivitykset ovatkin sittemmin korjanneet jälkimmäisen puutteista. Syksyllä 2018 päivitys lisäsi myös kolmannen uupuneen asian, ajoneuvotyyppien tunnistuksen mittaristonäytölle. Mittaristolla näkyi nyt myös 360 asteen havainnekuva liikenteestä (ks. kuva). Takakamera sekä sivukameroiden etäisyysarviot olivat tosin epäluotettavia. Ohittavat autot näyttivät usein törmäilevän ja takana olevat autot katoilivat.

Amerikasta kuului vieläkin kummempia. Siellä EAP sai myös *Navigate on Autopilot* -toiminnon (NoA), joka kykeni seuraamaan navigaattorin reittiä moottoritien rampilta toiselle sekä vaihtamaan navigaattorin perusteella jopa moottoritien kaistaa. Jälkimmäinen ominaisuus tosin julkaistiin al-

le jopa hieman ennen Waymo Onea. ZMP:n ratissa istuu aina turvakuljettaja.

Suomessakin tapahtuu. Sensible4 aloitti kuskittoman robottiminibussikokeilun Espoon Otaniemessä huhtikuussa. Sitä edelsi pikkupilotteja Renault Twizy -sähköautolla vuodesta 2017 alkaen. Bussin muotoili japanilainen Muji.

Raamatun maailma

Teknologiavalintojen rinnalle on muodostunut toinenkin itseajavien autojen jaottelu: robottitaksit ja autonomiset henkilöautot. Siinä missä Waymo, Sensible4 ja kumppanit tähtäävät joukkoliikenteen osaksi näkyvillä ja kalliilla sensoripattereillaan, Audin ja Teslan kaltaiset autonvalmistajat kehittävät itseajavia autoja, joita kuluttajat voivat ostaa ja jotka eivät ulkoisesti poikkea tavallisista autoista.

Tärkeä peluri näiden maailmojen välissä on israelilainen MobilEye (nykyisin osa Inteliä), joka kehittää konenäköpiirejä ajoavustimiin ja itseajaviin autoihin. MobilEyen EyeQ3-piiri löytyy sekä Audin Staupilotista että Teslan ensimmäisen sukupolven Autopilotista. Viime vuonna julkaistu EyeQ4-piiri on otettu käyttöön esimerkiksi BMW:n uusissa malleissa. Lisäksi MobilEye kehittää omaa robottitaksia Israelissa perustuen EyeQ4:ään.

MobilEyen jo yli kymmenen vuoden ikään ehtinyt EyeQ-piiriperhe sisältää uskottavan arsenaalin itseajamiseen tarvittavia toimintoja, kuten luotettavan konenäön, kartoituksen ja uutuutena jopa ajopäätöksenteon. Ne ovat osanneet vuosia monia sellaisia asioita, joita esimerkiksi Tesla on toteuttanut itserakentamaansa Autopilot 2:een vasta tänä vuonna. Valitettavasti autonvalmistajat ovat olleet laiskoja näiden ominaisuuksien käyttöönotossa.

Silti edistystäkin tapahtuu. Uusimmat kolmella tuulilasikameralla varustetut BMW:t, kuten BMW X5, sisältävät EyeQ4:n automaattisen kartoitustoiminnon (*REM Mapping, Road Experience Management*). Kaikessa hiljaisuudessa bemaristit ylläpitävät siis ajantasaista tarkkuuskarttaa maailmasta, kuten kaista- ja poikkeusjärjestelyistä, jota tulevaisuuden itseajavat autot voivat hyödyntää – tai olla hyödyntämättä, jos uskoo Muskia.

Vaikuttavin on kuitenkin MobilEyen robottitaksiprototyyppi, joka näyttää ulospäin tavalliselta Ford Fusionilta. Autoa on esitelty sekä Jerusalemin hullussa liikenteessä että muuten vaan hullussa CES:ssä. Se käyttää pelkkää konenäköä, mutta MobilEyen johtaja **Amnon Shashua** on kuuluisasti asettunut vastakarvaan Muskin kanssa suosittelemalla Lidaria ja radiotutkaa rinnalle turvallisuussyistä. Lisäkerros havainnointikykyyn onkin luvassa robottitaksin seuraavaan versioon.

Maailmanloppu

Tesla esitteli huhtikuun sijoittajatapahtumassaan uuden *FSD Computer* -nimisen autotietokoneen, jota on lupailtu aikanaan myös jälkiasennuksena Full Self-driving -varusteella varustettuihin vanhempiin Tesloihin. Sen ytimessä on kaksi Teslan itse kehittämää neuroverkkokiihdytintä, joista kirjoitin kolumnissani (Skrolli 2019.1). Tämä tietokone ja ohjelmistopäivitys yhdistettynä Tesla-autojen kahdeksaan kameraan on Muskin mukaan kaikki, mitä robottiauto tarvitsee.

Musk kertoi Teslalla olevan jopa suunnitelmissa poistaa nykyisistä autoista ratit ja asentaa tulppa tilalle, jos omistajat niin haluavat. Ajatus on, että Tesloja voi noin ensi vuodesta alkaen liittää *Tesla Network* -robottitaksiverkostoon, jossa ne tienaavat kymmeniä tuhansia dollareita vuodessa omistajalleen, sillä aikaa kun tämä pakertaa päivätyössä tai nukkuu. Mihin sitä rattia tarvitaankaan, kun Pole Positionkin jouduttiin poistamaan Tesloista (s. 80).

Itseajamisen eri leireillä on omat perustelunsa. Monisensoriuskovaiset muistuttavat, että eri antureilla on vahvuutensa: Kimpoilevat radiotutkasignaalit näkevät esineiden ”lävitse.” Aktiivisena valolähteenä Lidar näkee pilkkopimeässä ja on hyvä arvioimaan etäisyyksiä sekä vastaamaan kysymykseen ”eihän siellä ole mitään.” Joissain visioissa autoihin lisättäisiin mikrobolometrit tunnistamaan lämpöeroja, joskaan tällaista ei vakavissa projekteissa ole näkynyt.

Kameraa tarvitaan toki aina yksityiskohtaisempien asioiden tunnistamiseen ja liikennemerkkien lukemiseen. Toisella laidalla ajatellaankin, että kun ihminen ajaa silmillään havainnoiden, miksei autokin voisi. Erilaisten sensorien fuusioiminen lisää kustannuksia ja vaikeuttaa päätöksentekoa, kun antureilta tulee ristiriitaista tietoa. Karttatieto taas vanhenee nopeasti, eikä

kuun ehdotusmuodossa, joka täytyy kuitata vilkulla ja myöhemmin ratin heilautuksella. Moottoritieltä poistuminen tapahtui kuitenkin automaattisesti, kunhan olet rampin viereisellä kaistalla.

NoA ei toimi vielä kovinkaan hyvin, mutta *Enhanced Autopilot* alkoi saada nimelleen vastinetta.

Talvi on tulossa

Kuten Teslan ohjelmistopäivityksissä yleensäkin, ei askeleita eteenpäin, ellei paria myös taakse. Vuoden 2019 lähestyessä Autopilotin osana toimiva neuroverkkopohjainen sadetunnistin lakkasi minulla ja monilla toimimasta lähes kokonaan. Näihin aikoihin internet alkoi kuhisemaan toisestakin ilmiöstä: Autopilot-kuskit olivat alkaneet törmäillä moottoritielle pysähtyneisiin paloautoihin.

Paloautoihin törmäämistä tapahtui kuitenkin sekä uudella että vanhalla Autopilotilla. Osasyynsä lienee yhdysvaltalaisella tavalla pysäköidä paloauto liikenneonnettomuuden suojaksi puolelletoista kaistalle. Paikallaan seisova este on Autopilotin sokea piste, kuorrutuksen alla kun kyseessä on vieläkin vain kaistavahti ja mukautuva vakionopeudensäädin.

Tammikuussa 2019 Yhdysvalloista kantautui sanaa, että ensimmäiset Teslan omistajat olivat saaneet rahojaan Autopilot 2 -joukkokanteesta, joka oli sovittu vajaata vuotta aikaisemmin. 20–280 dollarin korvauksia järjestelmän viivästyksistä lupailtiin aikanaan Yhdysvaltain ulkopuolellekin, mutta tästä ei ole kuulunut mitään lehden mennessä painoon. Aikaa Tesla oli kuitenkin saanut ostettua – EAP:n automaattinen ohjaus kun pysytteli nyt jo varsin hyvän ja vain kohtuullisen pelottavan välillä.

Toinen kokonainen talvi kerättyjä kokemuksia vahvisti ensimmäisen talven havaintoni: kalifornialainen auto ei pidä

Lidar tarjoa radikaalisti kamerasta poikkeavaa näkökykyä, koska sekin toimii pitkälti näkyvän valon alueella.

Myös koneoppimiskysymys on jakautunut leireihin. Progressiivisimmat, kuten Tesla ja comma.ai, ovat esittäneet, että kuluttaja-autoista kerättävä iso data ja amatöörikuljettajien massan tukema syväoppiminen on ratkaisun avain. Tässä niin sanotussa *fleet learningissa* ne ovatkin edellä, vaikka keräävätkin autoista vain pieniä välähdyksiä tietoa. Tiedon käyttö varsinaisessa neuroverkkokoulutuksessa on tosin kysymysmerkki. Heillä on etulyöntiasema myös uusien toimintojen käyttöönotossa: autojen ohjelmistopäivityksiä voi tehdä etäältä, ja yrityksillä on hullunrohkeutta, jota vanhoilta puuttuu.

Konservatiivisen näkemyksen mukaan algoritmit ja perinteinen koneoppiminen ovat edelleen tehokkaita ja ennen kaikkea turvallisia ratkaisuja, jotka eivät edellytä kuluttajien apua. Syväoppimiselle on kasvava paikka kaikkien suunnitelmissa – sillä voidaan esimerkiksi ratkaista se sensorifuusio – mutta vanhan liiton mukaan ammattilaiset ja simulaattorit kouluttavat neuroverkkoja parhaiten.

Edelleenkään kukaan ei tosin tunnu vakavissaan kehittävän *end-to-end*-robottiautoa, jossa syväoppinut, kameroita katseleva neuroverkko ohjaisi sellaisenaan autoa. Kaikkien pelureiden ratkaisut ovat kokoelmia erilaisia tekniikoita ja painotuksia. Musk ylitti kuitenkin jopa itsensä vihjaamalla sijoittajapäivässä, että end-to-end-ratkaisu on lopulta Teslan tavoitteena.

Post scriptum

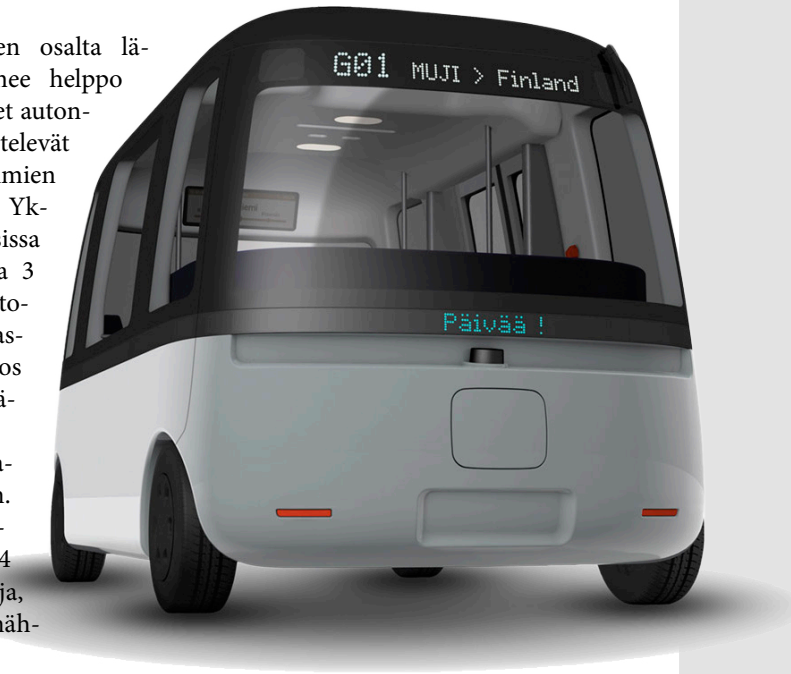
Monien toimijoiden osalta lähitulevaisuutta lienee helppo ennustaa. Perinteiset autonvalmistajat näpertelevät tason 2 ajoavustimien kanssa lähivuodet. Yksittäisissä tapauksissa he kokeilevat tasoa 3 jossain suuressa automaassa kuten Saksassa tai Japanissa – jos Saksassa, EU:n sisämarkkinasäädökset saattavat sallia saman Suomessakin. Useilla autonvalmistajilla on tason 4 robottiautoprojekteja, mutta niitä tuskin nähdään vuosiin.

Waymon ja Sensible4:n kaltaisia alueellisia robottitaksi- ja bussioperaatiota alkaa putkahdella eri puolille maailmaa. Kuluttajat eivät voi ostaa autoja, mutta todennäköisyydet päästä kyytiin kasvavat. Ajan myötä näemme muitakin ammattiliikenteen ratkaisuja, kuten itseajavia rekkakolonnia. Hamassa tulevaisuudessa nämä teknologiat saattavat päätyä lisenssillä myös kuluttaja-autoihin.

Jäljelle jäävät kaksi mustaa hevosta. Teslan osalta ennustaminen on kolikonheittoa. Joko sen lähestymistapa on valtavasti muita nopeampi tai se puhuu puuta heinää. En lähde arvailemaan. MobilEyellä ovat itseajamisen natsat ja aggressiivisuuskin kunnossa, mutta keksiikö se ratkaisua autonvalmistajien passiivisuuteen? Sääli, että Teslan ja MobilEyen sukset menivät taannoin ristiin.

Ensi kertaan! Palstalle voi kirjoittaa osoitteeseen janne@skrolli.fi.

Maksuttomat pdf-versiot yli vuoden vanhoista Skrolleista skrolli.fi:ssä: skrolli.fi/numerot



Gacha on suomalais-japanilainen autonominen minibussi. Se käyttää espoolaisen Sensible4:n robottiautotekniikkaa, joka hyödyntää kameroiden lisäksi neljää Lidaria ja kahdeksaa tutkaa sekä tarkkuuskarttoja. Gacha aloitti liikennöinnin Suomessa pääkaupunkiseudulla ja bussia tullaan näkemään tänä vuonna muuallakin. Sen lupaillaan toimivan myös talvella. (Kuva: Sensible4)

Suomen ilmastosta. Toisen sukupolven Autopilotin tutka on lämmittämättömän puskurin takana, joten EAP-matka katkeaa yllättävän ohueen lumikerrokseen Teslan keulassa.

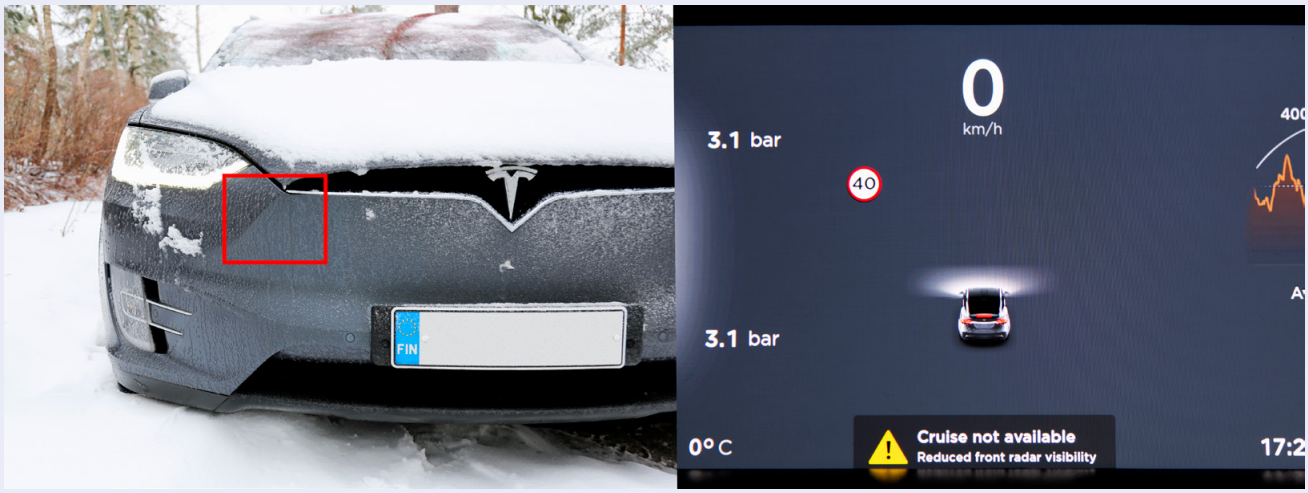
Silloin kun järjestelmä toimii talvella, saa kaistamerkintöjen kanssa olla erityisen tarkkana. Jos siis ylipäätänsä on päässyt jäätyneistä karmittomista ovista Teslaan sisään. Onneksi Teslan etälämmitys on hyvä. Model X:n kanssa ei ole sentään tarvinnut kärsiä maalinkulumisongelmasta talvihiekoilla, josta eräs suomalainen Model 3 -omistaja on puuhannut #teslapaintgate-valituskampanjaa.

Uskomaton puhallus

Vuoden 2019 alkupuolella Tesla uudisti ajoavustimiensa hinnoittelun. *Full Self-Driving* -ominaisuus palasi myyntiin, mutta vanhojen asiakkaiden jälkiasenteinen päivityshinta koko paketille tipahti noin viiden tonnin paikkeille. Hinta oli alkujaan Suomessa noin 8 800 euroa uutena ja jälkipäivitykset ovat maksaneet jopa yli kymppitonnin (hinnat ennen autoveroa).

Nyt hinta romahti puoleen, ennen kuin FSD:n täyteen hintaan ostaneille asiakkaille oli toimitettu ensimmäistäkään vain FSD-pakettiin kuuluvaa ominaisuutta. Musk oli kuuluisasti luvannut FSD-ominaisuuksia kevätkesälle 2017. Nyt elettiin maaliskuuta 2019. Tässä saikin *early adopter* -lisähinta jo ihan uuden merkityksen. Maksoit tuplasti, ilman että sait mitään.

Vuosisadan diili. Nähdäkseni vähemmästäkin on päässyt **Mikko Heinosen** legendaariselle *Ei näin!* -palstalle Skrollissa. Vanhan hinnan maksaneita asiakkaita lepytelläkseen Tesla lupasi pääsyn *Early Access Program* -ohjelmaan, jolla saattaa saada joitakin ohjelmistopäivityksiä etukäteen – vastineeksi siitä, että ei kirjoi-

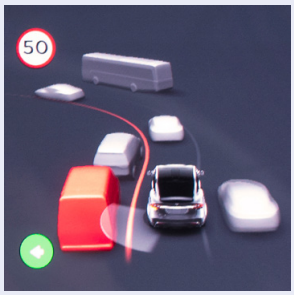


Kalifornia-auto Suomen talvessa. Tämä ohut kerros loskaa riittää sekoittamaan puskurin takana olevan tutkan (punaisessa neliössä) ja Autopilotin.

ta niistä esimerkiksi Skrollin *Ei näin!* -palstalle. Juuri kukaan ei tosin tunnu saaneen kutsuja päivitysohjelmaan.

Alattekin ehkä huomata tietyn kuvion toistuvan Teslan puheissa. Ne ovat puheita. Internetissä levisikin pian vitsi, että Teslan *Full Self-Drivingissa* sana *self* viittaa auton kuljettajaan, joka ajaa täysin itse. Tesla lopetti alennukset paria viikkoa myöhemmin, ilmeisesti kerättyään sopivasti käteistä kvartaaliraportille. Maanisista hinnanmuutoksista on kuitenkin tullut osa Tesla-arkea, koska hinnat ovat sittemmin muuttuneet taas – useamman kerran ja useampaan suuntaan.

Aivan itse ei sentään tarvinnut ajaa, kiitos EAP:n, joka jatkoi kypsymistään. Jotkut raportoivat vanhojen haamujarrutuksien paluusta, mutta minulla automaattiohjaus sekä kaistojen ja ympäröivän liikenteen tunnistus mittariston ruudulla toimivat jo varsin sulavasti. Siltojen alta päästiin ilman haamujarrutuksia ja ohjaus oli varmempaa. Parannus vuoden takaiseen oli selvä. Kaistaa vaihdettaessa kuolleessa kulmassa olevasta autosta tuli nyt myös visuaalinen varoitus, ja hieman myöhemmässä päivityksessä saatiin äänimerkkikin.



Muistutuksia tarkkaavaisuuden tarpeesta tuli silti tasaisen tappavaan tahtiin. Yhdysvalloissa Autopilot pääsi osalliseksi jo toiseen kuljettajan dekapitaatioon, kun Tesla Model 3 ajoi poikittaisen rekan alta. Kuljettaja on ajamisesta vastuussa, mutta on myös selvää, että Teslan kyky tunnistaa yllättäviä esteitä on erittäin rajallinen. Auton etututka on tässä varsin epäluotettava kumppani, eikä ultraäänisensoreista ole apua nopeassa liikenteessä.

Kaikkien estetyyppien opettaminen kameroille on iso työ. Lidar auttaisi.

Haamuja ja menninkäisiä

Olen ajanut kuudella tutkavakionopeudensäätimellä varustetulla autolla vuodesta 2006 alkaen. Niiden yleinen käytös on tuttua. Usein mukautuvat vakionopeudensäätimet reagoivat vain samansuuntaiseen, edessä ajavaan liikenteeseen – tutkasignaalista pysäköityjä kohteita on vaikea erottaa. Teslankin varhaiset haamujarrutukset perustuivat tutkasignaalin väärintulkintaan ja sopivat vanhoihin kokemuksiin, vaikka olivatkin pahempia kuin vanhoilla autoillani.

Huhtikuussa pääsin kokemaan uudentyyppisistä haamujarrutuksia. Ohjelmistopäivityksen jälkeen auto alkoi reagoimaan jarrulla myös vastaantulevaan ja poikittaiseen liikenteeseen, jos se vaikutti uhkaavalta. Vaikka tämä aiheutti mutkissa ajoittaista reagointia myös liikenteenjakajan takana turvallisesti tapahtuvaan liikenteeseen, se ei tuntunut kovin haitalliselta. Tesla oli selvästi alkanut luottamaan kameroihinsa muussakin kuin kaistaviivoissa ja edellä ajavissa autoissa.

Teslan sijoittajapäivässä tekoälyjohtaja **Andrej Karpathy** esitteli uutta neuroverkkoa, joka ennustaa kaistanvaihtoja auton edessä. Sain oletettavasti makua tästä Länsiväylällä viikkoa myöhemmin, kun Tesla alkoi yhtäkkiä jarruttamaan mustan Audin ohittaessa minua, vaikka kaistani oli tyhjä. Ennen kuin ehdin kirota Autopilotia, Audi leikkasi kaistalleni toistuvasti – ei paljoa, eikä vaihtaakseen kaistaa, mutta selvästi kaistaviivan ylitse. Koneälyn ennakointi tuntui osuneen kohdalleen.

Kevättalvinen ohjelmistopäivitys toi kaikessa hiljaisuudessa myös *Fleet Speed* -nimisen ominaisuuden. Toisen sukupolven Autopilot ei osaa lukea nopeusrajoitusmerkkejä, toisin kuin MobilEyen tekniikkaan perustuva Autopilot 1, joten se on määritellyt maksiminopeutensa karttatietokannan perusteella (pääkaupunkiseudulla edelleen usein noin 10 km/h liian korkeaksi).

Uusi ominaisuus kuitenkin seuraa muiden Teslojen ajamia nopeuksia ja muodostaa niiden perusteella sisäisen Fleet Speed -nopeusrajoituksen esimerkiksi moottoritierampeille. Näin auto hiljentää automaattisesti tähän piilo- tai pitäisikö sanoa pilvinopeuteen. Maailmalla tästä yllätyksestä on raivottu, mutta itseäni ominaisuus ei ole häirinnyt.

Samoihin aikoihin Tesla toi autoihinsa *Sentry Mode* -tilan. Se ei varsinaisesti ole osa Autopilotia, mutta se käyttää samaa tekniikkaa pysäköinnin aikana. Teslaa ympäröivät kamerat kuvaavat auton ympäristöä ja neuroverkko tulkitsee näkemäänsä. Jos joku esimerkiksi nojaa autoon, käynnistyy varoitustila ja tallennus. Mikäli ikkuna rikotaan, alkaa musiikki soida täysillä ja auton omistaja saa varoitusviestin mobiilisovellukseensa. Ainutlaatuinen ominaisuus tehdasvarusteeksi.

Sitten Sentry Moden kääntöpuoli: se ei varsinaisesti toimi varhaisemmissa Autopilot 2 -autoissa. Auto kyllä reagoi lähestyviin uhkiin näyttämällä auton näytöllä valvovaa Sentry Mode -silmää (HAL 9000) ja vilkuttelemalla ulkovalojaan villisti – se tekee sitä ”mukavasti” myös pelkälle sateelle tai viereiselle talolle – mutta muistitikulle tallentuu tyhjiä videotiedostoja. Sentry Mode toimii paremmin noin elokuusta 2017 alkaen valmistetuissa Tesloissa.

Takapajula Eurooppa

Yksi ongelma Autopilotin kehityksen seuraamisessa on, että usein ajamiseen liittyvät ominaisuudet julkaistaan ensin Teslan kotimarkkinoilla Yhdysvalloissa ja seuraavaksi Kanadassa. Esimerkiksi *Enhanced Summon* -etäkutsu parkkipaikalla uupuu edelleen eurooppalaisista Tesloista lehden mennessä painoon. Muistetaan, että vastaavia ominaisuuksia lupailtiin alkujaan jo vuodelle 2016 – osittain jopa Autopilotin ensimmäiselle sukupolvelle. Odotamme edelleen.

Perus-Summon meillä sentään on. Sekin on mainitsemisen arvoinen ominaisuus, vaikka vastaavia on muillakin valmistajilla. Summon mahdollistaa auton liikuttamisen kännykkäsovelluksella, eteen- tai taaksepäin. Mikäli auto on HomeLink-autotallinavaimella varustetussa autotallissa, Tesla osaa jopa avata autotallin oven. Tosin sen kerran, kun olisin itse tarvinnut Summonia, olin Helsingin Kampissa parkkihallissa maan alla. GPS-varmistus ei uskonut, että seison auton vieressä, joten auto ei liikahtanut Summonilla mihinkään.

Enhanced Summon ottaa askeleen eteenpäin ja mahdollistaa Teslan kutsumisen luokseen tai kartalta valittuun lähipaikkaan jopa 45 metrin päästä. Tällaista ominaisuutta ei ole muissa automerkeissä. Teslaan on pidettävä katsekontakti ja kutsunappia on pidettävä pohjassa, mutta YouTube-videoiden perusteella se tosiaan ajelee itsensä parkkipaikaltaan perille kohteeseen – kylläkin niin hitaasti, että tositilanteessa kanssa-autoilijat polttaisivat kääminsä. Parkkiraivoepidemiasta ei vielä ollut pelkoa edes Amerikassa, sillä – kuten tavallista – Tesla on markkinoinut uutuutta laajalti, mutta oikeasti se on ollut vasta pienen testiryhmän autoissa.

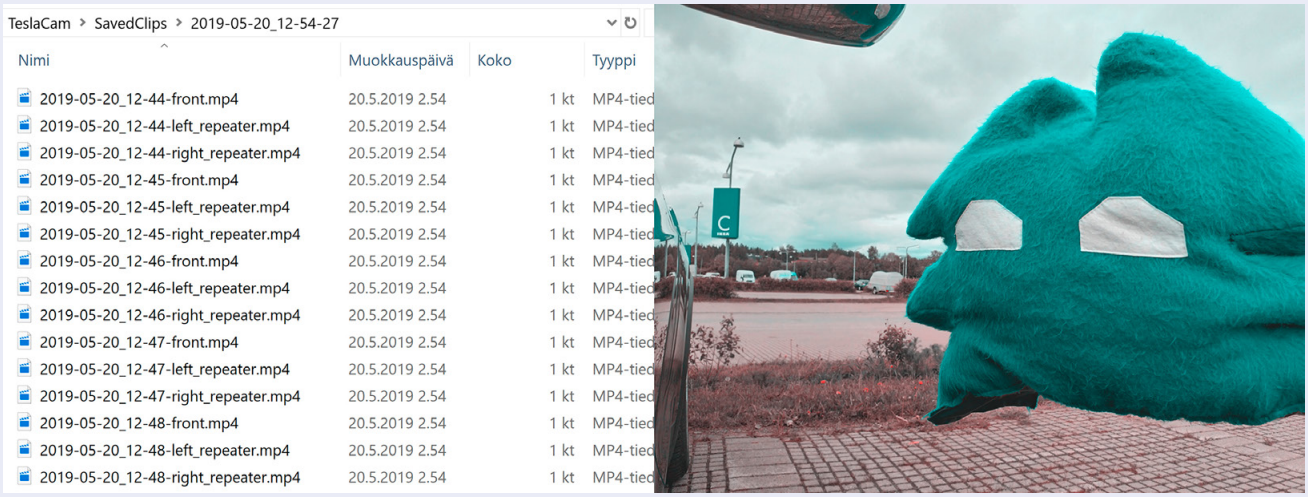
Edellä mainittu *Navigate on Autopilot* -toiminto saapui laajasti Eurooppaan tätä Skrollia viimeistellessämme, puoli vuotta Amerikan jälkeen – ei kuitenkaan vielä meille. Uusien säädösten vuoksi Tesla otti samalla taas takapakkia. Tesla rajoitti Autopilotin tekemiä jyrkkiä käännöksiä sekä muutti Autopilotilla ajettaessa tehtäviä automaattikaistanvaihtoja niin, että vilkkuvipua on näpäytettävä hetkeksi puoliväliin, jotta kaistanvaihto käynnistyy. Aiemmin ominaisuus on toiminut tavallisella vilkun kytkemisellä. Vilkkumuutoksen kanssa voi elää, mutta kääntösädemuutos on aivan kamala – auto pyrkii nyt tieltä ihan normaaleissa mutkissa.

Nämä eivät varsinaisesti ole EU- tai Eurooppa-säädöksiä, vaan perustuvat YK:n UN/ECE R79 -harmonisointisopimukseen. Ne voivat siis tulla voimaan muuallakin kuin Euroopassa. Vaatiikin aika paljon salapoliisityötä selvittää, mitkä ominaisuudet todellakin toimivat missäkin Tesloissa, koska salailevan Teslan tiedotus on painottunut ylimalkaiseen PR-hypeen, eikä tarkempia muutoslokeja ole saatavilla. Musk on usein tviitannut ominaisuuksien valmistumisesta kuukausia, jopa vuosia ennen kuin ne oikeasti ilmestyvät laajalti asiakkaiden autoihin missään, saati meillä Euroopassa.

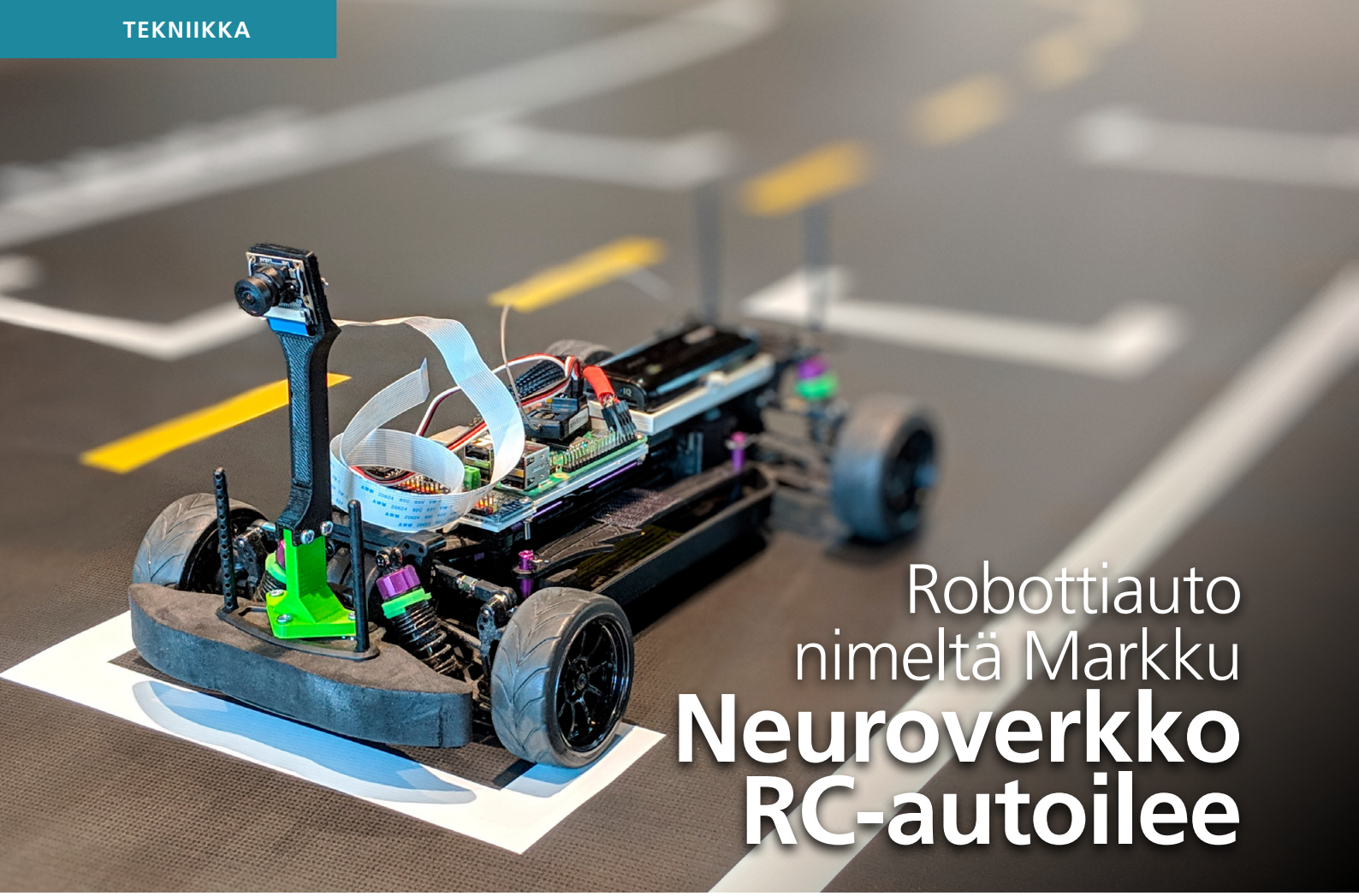
Teslan toisen sukupolven Autopilot täyttää kolme vuotta tänä vuonna, jolloin moni yritys- ja leasing-auto menee aikataulun mukaan vaihtoon. Teslan maailmanlaajuisen päivitystahdin vuoksi on mahdollista, että jotkut autot eivät ehdi saada ensimmäistäkään Autopilotin FSD-ominaisuutta ennen sitä, vaan auton mennessä vaihtoon Autopilot on jotakuinkin samanlainen kuin ensimmäisen sukupolven Autopilot oli vuonna 2015 – pienin puuttein ja pienin parannuksin.

Perinteiset autonvalmistajat ovat toki vieläkin hitaampia liikkeissään, mutta Tesla on ainoa, joka on kerännyt ennakkomaksun.

Nykyisin Tesla myy osaa EAP:n ominaisuuksista FSD-paketissa ja EAP:n myynti on lopetettu. EAP:n korvaa uusi perustoiminto Autopilot, johon kuuluu yksinkertainen automaattiohjaus ja -nopeudensäätö.



Aikainen Tesla-asiakas ei nappaa matoa, eikä mörköä. Sentry Mode kun ei toimi ennen elokuuta 2017 valmistetuissa Autopilot 2 -autoissa. Kuvassa on lavastus Teslan sivuvilkkukameran kuvakulmasta, kun mörkö lähestyy (kuvan ylänurkassa pilkistää auton sivupeili, Skrolli-trolli levitoi puolisen metriä ilmassa). Muistitikulle tallentuu kuitenkin tyhjää videota. Asiaa selvitelleet eivät ole kyenneet vahvistamaan, että kuvaa lähetettäisiin Teslan pilveenkään. Pääkameroiden värisuodatin on puna-mustavalkoinen, joten kuva näyttäytyisi tällaisena turkoosihtavana. Myöhemmissä Tesloissa suodatin on puna-sini-mustavalkoinen, jolloin kuva on kellertävä – niissä tallennus toimii.



Robottiauto nimeltä Markku Neuroverkko RC-autoilee

Itseajavia autoja on monen kokoisia. IT-firma Futuricen tiimi lupautui kertomaan Skrollin lukijoille, miten radio-ohjattava auto taipuu tekoälyn hallintaan.

Teksti: Lauri Anttila, Mikko Pohja
Kuvat: Lauri Anttila, Felipe Santana, Mikko Pohja

Tampellan perinteikkäässä tehdaskiinteistössä on rauhallista. Tammerkoski virtaa vieressä leppoisasti ja ratikkatyömaan äänet ovat hetkeksi hiljentyneet. Rakennuksen neljännestä kerroksesta kuuluu tyypillisiä driftausääniä ja Ace of Spadesia. RC-auto kiitää rataa ympäri eikä sen ohjaajaa näy missään. Siis, mitä? Kelataanpa hieman taaksepäin ja hypätään Futuricen Tampereen-toimiston arkeen.

”Kyllä me parempi tehdään”

Kaikki hyvät tarinat saavat alkunsa jostain. Meidän tapauksessamme taustalla oli perinteinen ”Kyllä me parempi tehdään” -mentaliteetti. Helsingin-toimipisteessämme oli syksyllä 2018 löydetty tietoa jostakin ”Donkey Carista” ja päätetty rakentaa sellainen. Ilman mitään ennakkotietoa aiheesta tai suunnitelmaa käytännön toteutuksesta noheva devaajamme Mikko ilmoitti välittömästi, että ”Voimme me kisan ottaa mutta tiedättehän miten siinä käy” (Tampere voittaa aina).

Pian meillä olikin käsissämme radio-ohjattavan auton runko (mallia Tamiya TT-02) sekä sekalainen kasa elektroniikkaa. Auto sai myös välittömästi lempinimen **Markku**, ja näin käynnistyi tarina, joka jatkuu edelleen. Jotta kohtalaisen kunnianhimoinen autoprojekti saatiin kunnolla käyntiin, oli edessä muutama ”haaste”, kuten konsulttipiireissä niitä ongelmia tupataan kutsua.

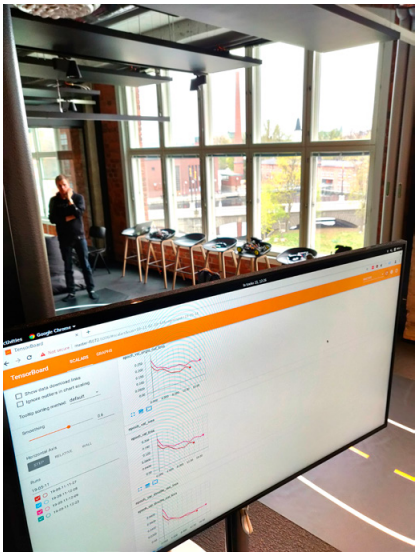
Ensimmäinen haasteista oli se, että kukaan hankkeeseen mukaan loikannut ei ollut juurikaan puuhastellut koneoppimisen saati autonomisten autojen kanssa. Ydinjoukko koostui webbi- ja mobiilidevaajista, jotka kuitenkin totesivat, että ”Ei kai se niin vaikeaa voi olla”. Projektista saatiin kyllä hyvä tarttumapinta kohti yrityksen tavoitteita tutustuttaa mahdollisimman suuri osa henkilöstöä koneälyn maailmaan.

Toinen ratkaistava kokonaisuus oli toki ajankäyttö sekä tilat, joissa autoa koulutettaisiin. Kokoluokka oli kuitenkin astetta Scalextricsia suurempi, joten ihan jokaisessa olohuoneessa saati toimistossa ei tila riittäisi radan rakentamiseen. Perinteisesti erilaiset ongelmat on kuitenkin saatu ratkaistua, kunhan on vain lähdetty kokeilemaan. Niin päätettiin tehdä nytkin.

Aikataulu tosin hieman venyi. Toimistojen väliset kilpailut piti käydä jo joulukuussa 2018, mutta erilaiset tekosyyt estivät tämän kisan tapahtumisen. Tämä puolestaan aiheutti harrastetoiminnan eskaloitumisen, josta syntyi samalla paljon hyvääkin.

Mikä ihmeen Donkey Car?

Ennen kuin sukelletaan syvemmälle projektimme pariin, on hyvä kerrata muutama perusasia liittyen Donkey Cariin. Projekti sai alkunsa vuonna 2016, kun **Will Roscoe** – nuori insinööri Oaklandista – päätti hakea paikkaa joukkoliikennetoimikunnasta. Hän oli tyytymätön San Franciscon joukkoliikenteen tulevaisuuteen, joka oli pitkälti suunniteltu raideliikenteelle. Hänen mielestään suunnitelmien pitäisi tähdätä käyttämään autonomisia busseja, jotka olisivat halvempia ja joustavampia. Roscoe päätti näyttää, että itseajavat autot ovat lähitulevaisuudessa mahdollinen ratkaisu, ja järjestää kokoontumisen aiheesta kiinnostuneelle joukolle ihmisiä.



Ensimmäisessä kokoontumisessa paikalla oli useita innostuneita rakentelijoita erilaisine robottiautoineen, joista Roscoen kiinnostus kohdistui kaikkein yksinkertaisimpaan autoon. **Adam Conwayn** rakentama auto perustui pelkästään Raspberry Pi -tietokoneeseen sekä siihen liitettyyn kameraan. Auton ohjauksesta oli vastuussa Googlen kehittämä avoimen lähdekoodin koneoppimiskirjasto Tensorflow.

Auto perustui mallioppimiseen, jossa neuroverkko opetetaan ajamaan samoin kuin sille näytetty ihmisen ajo. Kaksikko päätyi kehittämään autoa eteenpäin, ja nyt kolme vuotta myöhemmin Donkey Car on suosittu työkalu autonomisten robottiautojen parissa. Vuodesta 2016 alkaen Oaklandissa on järjestetty kuukausittainen kokoontuminen, joista kirjoitushetkellä viimeisimmässä paikalla oli yli 450 ihmistä ja kymmeniä autoja.

Projektin henkieläimeksi muodostui hieman erikoisesti aasi. Nimen takana on ajatus siitä, että on hyvä muistaa, kuinka tyhmiä ja itsepäisiä tekoälysovellukset tyypillisesti ovat. Aasi sopii kuvaan myös sen takia, että tavoitteena on kuitenkin luoda lapsiystävällinen ja hyödyllinen olento.

Entä mitä se aasi sitten on syönyt?

Rautapuoli autossa on hyvinkin yksinkertainen. Tavallisessa RC-autossa on akun ja moottorin lisäksi nopeussäädin (ESC) ja ohjausservo. Nämä on normaalisti kytketty radiovastaanottimeen, mutta tietokoneohjausta varten ne kytketään Raspberry Pi -tietokoneeseen. Raspberryn kaveriksi laitetaan kamera laajakulmalinssillä ja radiovastaanottimenkin käyttämää PWM-signaalia tuottamaan voidaan käyttää halpaa PCA9685-servo-ohjainta. Lopuksi lisätään kokonaisuuteen vielä USB-liitännällä varustettu virtapankki, jotta auton oman akun loppuminen ei rajoita Raspberryn käyttöä. Tällä peruspaketilla päästäänkin sitten jo asentelemaan ohjelmistoja ja kokeilemaan, miten mutkat taittuvat.

Toki rehellisyyden nimissä ostoslistalle päätynee helposti johtoja, SD-kortti, Bluetooth-ohjain, ynnä muuta pikkusälää, jotka helposti ensimmäistä tilausta tehdessä unohtuvat. Laitteiston puolesta homma on siis helppoa, halpaa ja hauskaa. Niin kauan, kun kaikki toimii.

Samaa voi sanoa valmiista Donkey Carin softasta. Asenna valmis image SD-kortille, aja muutama komento ja kaikki toimii. Ainakin ohjeissa ja teoriassa. Todellisuudessa Donkey Carin käyttöönottokokemus vastaa montaa muuta harrasteprojektina syntynyttä avoimen lähdekoodin projektia. Koodia pääsee ja joutuu lukemaan, versioristiriitoja tulee vastaan ja apua joutuu kysymään yhteisöltä. Kuitenkin niiden ensihetkien koittaessa, kun auto liikahtaa ohjaimen vaikutuksesta, lähtee omistajaansa karkuun AI:n voimalla ja lopulta pysyttelee viivojen välissä puolikin kierrosta, tunnelma nousee välittömästi kattoon.

Logiikkana auton softa on hyvin yksinkertainen: kaiken ajurina toimii vuorontaja, joka kyselee silmukassa sille rekisteröidyiltä palasilta niiden ulostuloarvot, antaen samalla niille niiden pyytämät sisääntulojen arvot. Esimerkiksi käsin ajaessa ”ohjain”-osalta saadaan renkaiden ja kaasupolkimen asento, joka annetaan fyysistä laitteistoa ohjaaville osille sekä ”tallennus”-osalle. Tätä jatketaan joka kierroksella, kunnes ajaminen lopetetaan. Näitä osia on helppo tehdä myös itse lisää, kuten me teimme lisättyämme autoon USB-kaiuttimen. ”Subwoofer”-niminen osamme ottaa sisääntulona ajotilan ja kehittämämme hätäjarrun tilan. Näitä käyttäen se soittaa tilaan liittyvää musiikkia (Ace of Spades, Knight Rider theme…) ja hätäjarrutuksen tapahtuessa kiljaisee hirmuisasti.

Donkey Car vs. AWS Deepracer

Amazon julkaisi oman roboautonsa marraskuussa 2018. Kyseessä on nopeasti vilkaistuna hyvin samankaltainen, mutta kaupallisempi avaimet käteen -versio Donkey Caristä, mutta konepellin alle katsottaessa eroja on melkoisesti. Miten nämä kaksi sitten eroavat toisistaan? Molemmat ovat pieniä (Deepracer 1:18, Markku 1:10) ja niissä on kamera sekä tietokone (Donkey Carissa Raspberry Pi, Deepracerissa Intel Atomiin pohjautuva prosessori).

Deepracer on Amazonin *reinforcement learningiin* nojaava alusta, kun taas Donkey Car pohjautuu *supervised learningiin* (tosin yhteisö on jo kehittänyt Donkey Cariin *reinforcement learningin* hoitavan osuuden). Deepracer siis tavallaan kouluttaa itse itseään, siinä missä Donkey Caria puolestaan kouluttavat ihmiset. Deepracer tarvitsee AWS:n pilviympäristön toimiakseen, mutta Donkey Caria voi-

daan käyttää joko off-line tai pilvessä. Donkey Car on myös täysin avoimen lähdekoodin tuotos, Deepracer taas täysin suljettu ympäristö.

Menemättä kaikkiin yksityiskohtiin voidaan sanoa, että Donkey Car tarjoaa joustavamman ympäristön laajentaa perusmallin ominaisuuksia, kun taas Deepracer keskittyy tarjoamaan standardin alustan kokeiluille. Standardoinnista vähät välittävälle näpertelijälle ja uuden oppijalle Donkey Car tarjoaa siis huomattavasti enemmän mahdollisuuksia.

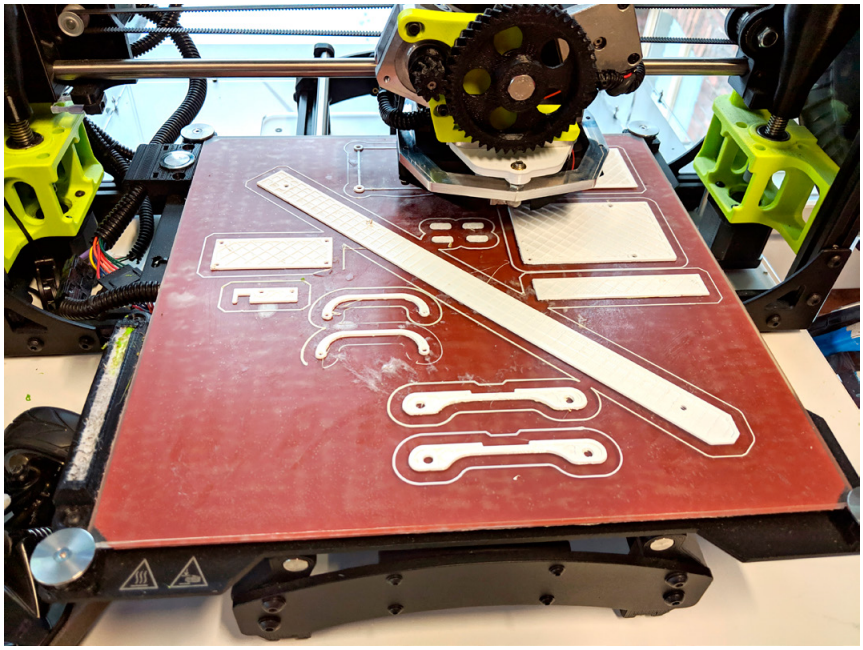
Kuinka konekuski koulutetaan?

Kunhan auto on koottu ja Donkey Car -image ladattu autoon, onkin jo *melkein* aika hypätä kouluttajan penkille ja alkaa opettamaan autoa ajamaan.

Donkey Car toimii siis *supervised learning* -periaatteella, eli se yrittää imitoida ”näkemäänsä”. Autolla ajetaan koulutuskierroksia, joiden aikana auto kerää dataa eli kuvia kamerasta ja ohjauskulmista. Näiden avulla voidaan omalla läppärillä tai pilvessä kouluttaa malli, joka sitten ladataan autoon. Mikäli kaikki sujuu kuin bensalenkkarien jumala on tarkoittanut, auto alkaa ajamaan omillaan.

Huomioitavaa on, että autoon ei erikseen ohjelmoida mitään radan reunaviivojen, liikennemerkkien tai muiden tunnettujen objektien tunnistusta, kuten monissa henkilöautoista tutuissa avustimissa. Näiden oppiminen perustuu puhtaasti siihen, mitä dataa sille annetaan. Meidän tapauksessamme opetimme ensin vahingossa automme tulkitsemaan toimistomme katossa olevia ilmastointiputkia merkityksellisinä elementteinä, ja varsinainen rata oli sille toissijainen kiinnostuksen kohde. Oppikokemusten myötä olemme muun muassa säätäneet auton kameran kulmaa ja leikanneet osia sen tallentamasta materiaalista saaden kuvaan kapeampaa sisältöä, jota oikeasti haluamme.

Mitä enemmän koulutusmateriaalia on tarjolla, sitä parempia malleja voidaan saavuttaa. Koska mittakaava on autossamme hieman erilainen kuin vaikkapa teollisuudessa, ei puhuta kuitenkaan aivan mahdottomasta datamassasta, jotta päästään riittävän hyviin tuloksiin. Käytännössä datan keruu meidän tapauksessamme tarkoittaa, että ihmisten on ajettava autolla ensin ratakierroksia toisensa perään ja tuotettava riittävä määrä materiaalia. Tarkoitusta varten käytimme hyväksemme muun muassa toimistollamme vierailevia opiskelijaryhmiä, jotta kuskia voitiin tehokkaasti vaihtaa. Kouluttaminen näin manuaalisesti oli luonnollisesti hyvin hidasta. Kumpa vain olisi jokin toinen keino...



Paisutellaanpa hieman dataa

Onneksi sellainen keino on olemassa ja sen nimi on *data augmentation*. Tämä tarkoittaa sitä, että pienestäkin datamassasta saadaan luotua paljon toimivaa koneälyn koulutusmateriaalia muokkaamalla ja monistamalla tallennettua tietoa. Esimerkiksi kameran kuvat ja ohjauskulmat voidaan monistaa peilikuviksi, jolloin materiaalia saadaan heti kaksinkertainen määrä.

Toimistollamme on lisäksi hyvin haastavat valaistusolosuhteet, sillä aurinko paistaa eri suunnista pitkin päivää ja lattiasta tulevat heijastukset tekevät radan tulkinnasta hyvin hankalaa (koska koulutusmateriaalia ei pystytä käytännön syistä nauhoittamaan kaikkien vuorokaudenaikojen ja sääolosuhteiden mukaan). Tätä ongelmaa pystytään ratkomaan muun muassa tuottamalla kuvista sellaisia versioita, joihin on lisätty keinotekoisia varjoja ja vaihtelevia kirkkauksia.

Pienelläkin määrällä koulutusdataa pääsee siis pitkälle monistamalla ja muokkaamalla sitä. Tämä säästää paitsi aikaa, myös kouluttajien hermoja.

Osaksi globaalia kehittäjäyhteisöä

Elektroniikan sovittamiseksi autoon piti eri komponenteille suunnitella ja toteuttaa jonkinlainen runko, joka kiinnitettäisiin auton alustaan. Alkuperäinen Donkey Car -design ei miellyttänyt tiimin esteettistä silmää, ja se todettiin myös aerodynaamisilta ominaisuuksiltaan riittämättömäksi edessä olevaan huipputason kilpailuun.

Tämän seurauksena ei ollut muuta vaihtoehtoa kuin laittaa 3D-softa (valintamme on Fusion360) tulille ja alkaa mallintamaan omaa koria. Useiden iteraatioiden jälkeen päädyimme nykyiseen, modulaariseen ”Chilicorn Rail” -konseptiin. Sen keskeinen elementti on *kisko*, johon voidaan kiinnittää erilaisia osia kameratolpasta Raspberry Pi’hin ja kannettavaan virtalähteeseen. Malleja on helppo muokata sovelluksessa ja tulostaa uudet iteraatiot käytännön kokeita varten toimiston 3D-tulostimella.

Autoa on rakenneltu työpäivien jälkeen hyödyntäen Futuricen ohjelmaa, joka kannustaa avoimen lähdekoodin sovellusten tuottamiseen ja jakamiseen, mutta myös työajalla projektien välissä. Ei voi kuin hattua nostaa työyhteisölle, jossa autoinnostus tarttui nopeasti ja siihen sijoitettu aika nähtiin selkeästi enemmän positiivisena kuin negatiivisena asiana. Oli luonnollista, että autoprojektimme tuotokset päätyisivät tavalla tai toisella avoimeen levitykseen. Ja näin tapahtuikin.

Kaikki mallit on julkaistu GitHubissa avoimen lähdekoodin hengessä il-

maiseksi, ja ne ovatkin lähteneet leviämään hyvin ympäri maailmaa Donkey Carin Slack-yhteisön välityksellä. Japanissa asti on jo tulostettu mallejamme, ja tätä kirjoittaessamme olemme saaneet Hong Kongista lahjaksi niin sanotun virallisen Donkey Carin HSP 94186 -rungon, jolle olemme myös julkaisseet oman version modulaarisesta kiskoratkaisustamme. Onpahan ensimmäinen pull request saatu myös kiskon sovittamiseksi meille vieraan RC-automallin alustalle.

Lisäksi auton rakentamisesta ja sen perusperiaatteista on koottu tietoa markku.ai-blogiin, joka saavutti vahingossa suosiota myös Hacker News -palvelussa tuoden iloisen kävijäpiikin sivustolle. Keskimääräinen liikenne sivustolla on vakiintunut mukaviin lukemiin ja tieto selvästi leviää maailmalle!

Koska tekijätiimillä ei muutenkaan ole kiireitä työn ja perhe-elämän yhteensovittamisessa, päätimme tottakai myös perustaa Meetup-yhteisön itseajavien autojen ympärille (www.meetup.com/Finland-Self-Driving-Cars-Meetup). Vastaavat meetupit keräävät sankoin joukoin harrastajia etenkin Yhdysvalloissa (ks. esim. www.meetup.com/DIYRobocars). Koska sellaista ei vielä Suomesta löytynyt, koimme velvollisuudeksemme korjata asian mahdollisimman nopeasti!

Tämän lehden julkaisuhetkellä toimintaa on saatu käynnisteltyä Tampereen suunnalla, mutta toivomme tapahtumien leviävän muuallekin. Ensimmäisessä meetupissa oli paikalla jo ainakin kuusi tiimiä omine autoineen muun muassa Aalto-yliopistolta, Tampereen yliopistolta ja Valmet Technologiesilta. Toisistaan eroavia teknisiä ratkaisuja oli mukava päästä ihmettelemään yhdessä paikassa ja ottamaan tuntumaa ajamiseen rata-olosuhteissa!

Mitä seuraavaksi?

Tätä juttua kirjoittaessamme on työlistalla jo paljon uusia kehityskokonaisuuksia. Auton ympärille on rakentumassa enemmänkin kilpailuja ja työpajoja, joissa koneoppimiseen pääsee tutustumaan miellyttävän kompaktissa ja ymmärrettävässä paketissa. Lisäksi, vaikka tavoite on pitää tekninen puoli mahdollisimman yksinkertaisena ja edullisena, testaamme myös uusia komponentteja ja tutkimme miten ne toimivat RC-autojen kontekstissa. Tällaisia ovat esim Nvidian Jetson Nano -tietokoneen käyttö Raspberryn sijaan sekä Intel Realsense T265 -kameran hyödyntäminen auton sijainnin seurannassa (ns. V-SLAMin hyödyntäminen).

Parhaiten Markun tarinasta saa irti seuraamalla Twitter-tiliä @maximum_aittack. Pyrimme jakamaan sen kautta kaiken kertomisen arvoisen erilaisista teknisistä kokeiluistamme. Valtaosa sisällöstä päätyy myös markku.ai-saitille, mutta Twitterin kautta välittyy myös paljon sellaista materiaalia, josta emme kirjoita pitempiä blogautuksia.

Kuinka pääsen alkuun?

Olemme koonneet omia kokemuksiamme markku.ai-blogiin. Lisäksi tiimin jäsenet Mikko ja Joose kävivät taannoin turisemassa projektista myös Webbidevaus.fi-podcastissa (webbidevaus.fi/16). Donkey Carin Slack-yhteisö on myös aktiivinen ja avulias.

RC-autojen saatavuus ja eri myymälöiden verkkokaupat ovat hyvinkin, noh, värikkäät. Varaudu käyttämään jonkin verran aikaan sopivien komponenttien löytämiseen.

Kuinka vaikeaa se voi olla?

PIENTÄ PDF-MUOKKAUSTA

PDF-tiedostot ovat mukavaa luettavaa, mutta tuskaista muokattavaa.

Teksti ja kuvat: Janne Sirén

Adoben vuonna 1993 julkaiseman PDF-tiedostomuodon (*Portable Document Format*) kauneus on siinä, että tiedostoja voi katsella käytännössä identtisinä ympäristöstä riippumatta. Sen pohjalla on PostScript-tulostusformaatti, joka kehitettiin 1980-luvun alussa kuvaamaan tulostettava sivu tulostinta varten.

PDF-tiedostoja ei täten varsinaisesti suunniteltu muokattaviksi, joten tiedostomuoto tarjoaa myös jonkin verran pysyvyyttä. Joskus tulee kuitenkin tarve tehdä PDF-tiedostoon pieni korjaus. Esimerkiksi sivujen ja kuvien kopioiminen PDF-tiedostosta toiseen käykin helposti melkein millä tahansa PDF-työkalulla. Tekstiä voi myös ainakin teoriassa muokata.

Kokeilimme formaatin muokattavuutta vaihtamalla yhden nimen ja numeron erääseen Skrollin PDF-tiedostoon. Kuinka vaikeaa se voi olla?

Virallinen vaihtoehto

Päivitimme taas vuodenvaihteessa Skrollin tehtävien jakoa. Tähän liittyen tarkoituksena oli vaihtaa Skrollin PDF-mediakorttiin **Toni Kuokkanen** toiseen rooliin siirtyneen **Jari Jaannon** paikalle. Mediakortti on media-alalla käytetty esite, joka kertoo median mainoshinnat ja -muodot. Pikkujuttu siis. Yleensä pullautamme vain Adobe InDesign -taitto-ohjelmasta mediakortin uuden version, mutta eikö näin pienen muutoksen voisi tehdä helpomminkin? Tällaisesta tilanteesta lähdettiin:

* Päivämäärät saattavat vielä muuttua.

Lisätietoja

Mediamyynti: **Jari Jaanto**
puh: **+358 40** [redacted]
email: **myynti@skrolli.fi**

Maksullinen Adobe Acrobat Pro DC 2019 (ei siis maksuton Acrobat Reader) on usein satunnaisen PDF-käsittelijän ykkösvaihtoehto, onhan se PDF-tiedostonmuodon kehittäjän ”virallinen ratkaisu” ja löytyy esimerkiksi Adoben Creative Cloud -palvelusta (abobe.com). Acrobatin Työkalut-valikosta (*Tools*) löytyvällä Muokkaa PDF-tiedostoa -toiminnolla (*Edit PDF*) voikin muokata mitä tahansa lukitsematonta PDF-tiedostoa. Tosin jos tiedosto on lukittu, voi joutua kokeilemaan kolmannen osapuolen työkaluja.

Mediakorttiamme ei ollut lukittu, joten pääsimme eteenpäin. Valitettavasti muutos kuitenkin rivittää muokkaamamme kohdan ja sen ympäristön väkisin uusiksi:

* Päivämäärät saattavat vielä muuttua.

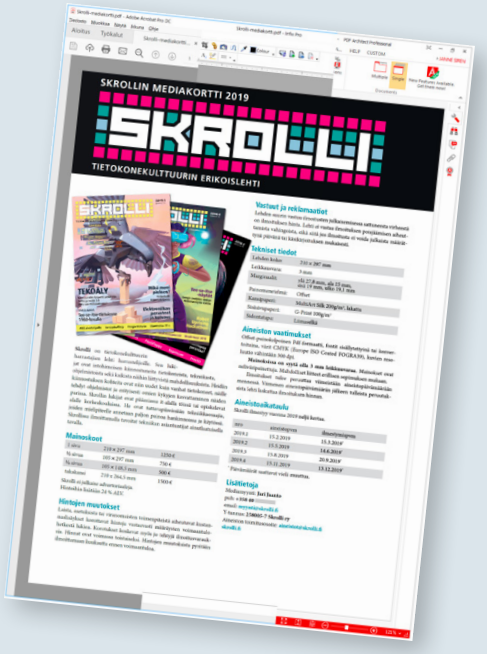
Lisätietoja

Mediamyynti: **Toni Kuokkanen**
puh: **+358 40** [redacted]
email: **myynti@skrolli. i**

TIedoston korjaaminen käsin osoittautuu vaikeaksi, ja tarkka silmä huomaa eroja myös kirjainväleissä. Lisäksi yksi i-kirjainkin katoaa jonnekin, jos puh:-alkuisen rivin korvaa uudella rivillä. Tämä vaikuttaisikin olevan aika tyypillinen PDF-muokkauksen epäonnistuminen Acrobatissa, kun PDF-dokumentin lähteenä on Adobe InDesign -taitto-ohjelma.

Työkalun ja fontin vaihto

PDF-muokkausohjelmia on pilvin pimein, joten seuraavaksi kokeilimme kahta Windows-ohjelmaa: PDF Architect (pdfforge.org) ja PDFescape (pdfescape.com). Voisi lähes vannoa, että kyseessä on sama ohjelma. Molemmissa iski myös sama ongelma: muutos ei nyt vaikuttanut riveihin, mutta useista yrityksistä huolimatta muokattu teksti pakottautui Times New Roman -kirjasimelle, vaikka tekstissä käytetty Minion Pro -fontti oli asennettuna tietokoneelle. Muutos saatiin siis tehtyä, mutta päivitetyn tekstin kirjasin oli nyt pielessä:



* Päivämäärät saattavat vielä muuttua.

Lisätietoja

Mediamyynti: **Toni Kuokkanen**
puh: **+358 40** [redacted]
email: **myynti@skrolli.fi**

Tässä vaiheessa heräsi tietysti kysymys, eikö ne väärät kirjasimet voisi vain vaihtaa oikeiksi jollain työkalulla. Yllättäen esimerkiksi Acrobat ei tarjoa tällaista toimintoa, mutta ainakin Infix Pro -ohjelmasta (iceni.com) sellainen löytyy (*Edit / Find & Replace / Fonts*). Valitettavasti meidän tapauksessamme tuloksena oli kirjasimen vaihtumisen lisäksi myös rivityksen vieläkin pahempi hajoaminen:

* Päivämäärät saattavat vielä muuttua.

Lisätietoja

Mediamyynti: **Toni** puh: **+ 3 5 8** [redacted] **40**
myynti@skrolli.fi

Hieman tekstiä muuttamalla tämä hajoaminen korjaantuisi, joten kirjasimen korvaamistoiminto sinänsä toimii. Riippuu ihan sivun koodauksesta, hajoaako rakenne vai ei, eikä tämä valitettavasti ole etukäteen läpinäkyvää käyttäjälle. Harjaantuneella silmällä jotain voi päätellä siitä, millaisiksi muokkauslaatikoiksi PDF kulloinkin jakaantuu muokkausnäkymässä – tällä on monesti vaikutusta muokkausten onnistumiseen.

Poistetaan upotettu fontti

Mitäpä jos vain poistaisimme sen ongelmallisen upotetun kirjasimen? Hyvänä uutisena upotetun kirjasimen voi tosiaan usein poistaa PDF-tiedostosta, esimerkiksi Acrobatin Optimize PDF -työkalulla (*Työkalut / Optimoi PDF /*

Lisäoptimointi / Kirjasimet eli *Tools / Optimize PDF / Advanced Optimization / Fonts*). Valitettavasti PDF-tiedostoilla on ajoittain myös taipumus haalia sisäänsä upotettuja kirjasimia, halusi tätä tai ei.

Acrobatin Optimize PDF -toiminto ei aina näytä tai tarjoa mahdollisuutta poistaa kaikkia upotettuja kirjasimia. Aina poistotoiminto ei myöskään poista upotettuja kirjasimia, vaikka siltä näyttäisi. PDF-dokumentin kirjasinten todellisen tilan ja analyysin PDF-tiedoston rakenteesta saat näkyviin esimerkiksi Acrobatin Preflight-toiminnolla (*Työkalut / Tulosta tuotanto / Preflight / Listaa mahdolliset kirjasinongelmat* eli *Tools / Print Production / Preflight / List potential font problems*).

Tässä tapauksessa PDF Architectin tuottama Times New Roman kirjasin ei ollut poistettavien kirjasinten listalla lainkaan, vaikka se upotettuna kirjasimena Preflightissa näkyykin. Vastaus artikkelin alun kysymykseemme alkoi jo valjeta: tämä voi olla aika vaikeaa…

Entäpä tiedoston muokkaaminen käsin? Onnistuisiko kirjasimen vaihtaminen sitä kautta tyylipuhtaasti?

Heksaeditori esiin

Hieman ilkikurisesti etsimme ensin heksaeditorilla (esim. *Hex Editor Neo*) alkuperäistiedostosta merkkijonon, jossa on oikean fontin nimi (tässä MinionPro-Bold) ja korvasimme sillä muokatun tiedoston väärät fontin nimet (tässä TimesNewRomanPS-BoldMT). Fontti löytyy lisäksi FontFamily-kohdasta kirjasinperheen nimellä, joten korvasimme myös sen. Mallia kirjasimen nimen esitystapaan voi hakea esimerkiksi muista PDF-tiedostoista.

Näin fontti vaihtuikin teoriassa. Acrobat näytti nyt tekstiä osoittaessa uuden kirjasimen nimeä, mutta ruudulle piirtyi edelleen väärä fontti. Syynä oli, että vaihdettava kirjasin oli upotettu PDF-tiedostoon, eikä sitä valitettavasti saanut poistettua Acrobatin kautta. Kirjasimen vaihtamiseksi on siis upottauduttava PDF:n rakenteeseen harkitummin.

Pakkaamaton PDF-tiedosto koostuu nipusta selväkielisiä oliomäärityksiä, jotka tunnistaa rakenteesta XX 0 obj.<< >>.endobj, jossa XX on olion tunnistenumero ja pisteet rivinvaihtoja (heksaeditorit näyttävät ne binäärimerkkien tapaan pisteinä). PDF-formaatti tarjoaa useita eri tapoja fonttien kuvaamiseen, kuten FontFile-, FontFile2- ja FontFile3-viittaukset, joiden käyttö riippuu kirjasimen tyypistä. Etsimällä näitä sanoja PDF-tiedostoista nähdään, miten niiden perässä viitataan kirjasimen olion tunnistenumeroon.

Tässä välissä on tosin todettava, että kirjasinmääritykset eivät aina esiinny PDF-tiedostoissa selväkielisenä. Syynä ovat PDF-versiossa 1.5 (Acrobat 6.0 alkaen) ilmestyneet oliovirrat (*object streams*), jotka mahdollistavat oliokuvausten lyhennetyn esitystavan. Niitä kutsutaan myös pakatuiksi olioiksi (*compressed objects*), sillä oliovirtoihin voidaan lisäksi soveltaa kompressiota. Nämä virrat tunnistaa PDF-dokumentista löytyvästä ObjStm-määreestä.

Pakatun PDF:n rakennetta voi tarkastella myös esimerkiksi Acrobatin Preflight-toiminnolla (*Työkalut / Tulosta tuotanto / Preflight / Asetukset / Selaa PDF-tiedoston sisäistä rakennetta* eli *Tools / Print Production / Preflight / Options / Browse Internal PDF Structure*) tai useilla muilla PDF-selaimilla, esimerkiksi O2:n PDFXplorerilla, Windjackin PDF Canopenerilla tai Winkingin PDF Analyzerilla, jotka purkavat kompression selkokielisempään muotoon.

Jos fonttimääritykset on kääritty oliovirtojen sisään ja pakattu binäärimuotoon, heksaeditorista ei ole juurikaan iloa. Tällöin PDF-tiedoston voi kuitenkin muuntaa pakkaamattomaan 1.4-versioon muokkaamista varten. Tämä onnistuu esimerkiksi Adobe Acrobatilla tallentamalla dokumentti muotoon *Acrobat 5.0 ja uudemmat* eli *Acrobat 5.0 or later* (*Tallenna nimellä / Muoto: Adoben PDF-tiedostot, optimoidut / Asetukset* tai *Save As / Save as type: Adobe PDF Files, Optimized / Settings*).

Vanhan koulun vaihto

Reilun parin vuosikymmenen aikana PDF-standardiin on kertynyt lukuisia tapoja kuvata kirjasimia. Fontin koodaustapa riippuu myös kirjasimen tyypistä. Näistä eniten keskustelua lienee herättänyt niin kutsuttu CID-koodaus, joka on valunut alkujaan aasialaisista merkistöistä myös länsimaisten kirjasinten käyttöön. CID-koodatut fontit ovat ilmaisuvoimaisia, mutta koodaukseltaan melkoisia mörköjä CMap-merkkikarttoineen. Riippuu siis täysin sekä kirjasimen tyypistä että valitusta esitystavasta, millaisena kirjasimet PDF:ssä näyttäytyvät.

Joskus muutokset ovat tehtävissä helpostikin. Esimerkkitapauksessamme PDF Architectin luomassa tiedostossa Times New Roman -kirjasimen ilmoitettiin FontFile2-määreellä löytyvän oliosta 38. Kun vaihdoimme PDF-tiedostoon viittauksen 38 tilalle toisen fonttiolion viittauksen 55 (samaan tapaan FontFile2-määrellä upotettu Play-kirjasin) kävi näin:

```
39 0 obj.<</Asce
nt 891/CapHeight
 891/Descent -21
6/Flags 6/FontBB
ox[-568.359375 -
306.640625 2045.
898438 1039.5507
81]/FontFamily(T
imes New Roman)/
FontFile2 38 0 R
/FontName/BZTZRK
+TimesNewRomanPS
MT/ItalicAngle 0
/StemV 1000/Type
/FontDescriptor>
>.endobj.40 0 ob
```

→

```
39 0 obj.<</Asce
nt 891/CapHeight
 891/Descent -21
6/Flags 6/FontBB
ox[-568.359375 -
306.640625 2045.
898438 1039.5507
81]/FontFamily(T
imes New Roman)/
FontFile2 55 0 R
/FontName/BZTZRK
+TimesNewRomanPS
MT/ItalicAngle 0
/StemV 1000/Type
/FontDescriptor>
>.endobj.40 0 ob
```

* Päivämäärät saattavat vielä muuttua.

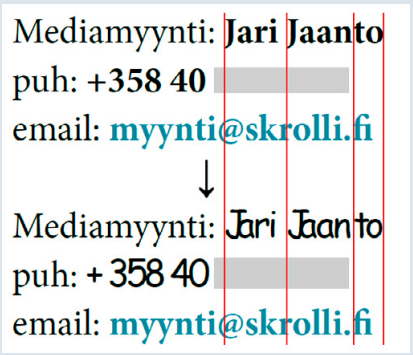
Lisätietoja

Mediamyynti: Toni uokk nen
puh:
email: **myynti@skrolli.fi**

Play-kirjasin oli upotettuna PDF-mediakorttiin historian jäänteenä. Fonttia käytettiin Skrollin kannessa ja logon yhteydessä ennen lehden ensimmäistä ulkoasu-uudistusta, numeroissa 2012.0–2014.3, joten siitä oli upotettuna vain sanojen ”Tietokonekulttuurin erikoislehti” kirjaimet. Kikka vaihtaakin kirjasimen, mutta koska vain osa siitä on mukana, katoaa puhelinnumero ja osa Toni Kuokkasen nimestä.

PDF-tiedostoon voisi toki olion 55 paikalle kopioida kokonaisen kirjasimen toisesta mallitiedostosta. Times New Romanin vaihto Minion Pro:ksi (PostScript Type 1 -kirjasin) ei kuitenkaan olisi enää tällainen kätevä pikkutuunaus, koska Type 1 -fonttien koodaus poikkeaa edellä mainituista. Esimerkiksi Subtype-määreen tulee vastata kirjasimen tyyppiä ja muutakin tarvitaan.

Entä miltä näyttäisi, jos upotetut kirjasimet poistetaan ja sen jälkeen kirjataan sopivat fonttien nimet heksaeditorilla paikoilleen? Ei kovin kauniilta, kuten alla oleva vaihdos Minion Pro Bold -fontista Comic Sansiin osoittaa. PDF-tiedostoihin nimittäin koodataan merkkien välitys itse kirjasimesta erillään. Näin ollen kirjasin kyllä vaihtuu, mutta kirjainten sijainti pysyy pitkälti alkuperäisenä:



PDF-tiedoston korjaaminen käsin ei siis yleensä ole realistinen tie, koska koodauksen saattaminen oikeaoppiseksi käy työstä. PDF-tiedosto on kuitenkin muokattavissa heksaeditorillakin, jos tietää mitä tekee. Mikäli innostuit, PDF-spesifikaatiot löytyvät artikkelin laatikon kautta.

Syvyyspommit esiin

Syy, miksi PDF Architectilla oli vaikeuksia alkuperäisen PDF-tiedoston kanssa, liittyi osittain upotettuun Minion Pro -kirjasimeen. Syystä tai toisesta fontin hakeminen sekä dokumentista että käyttöjärjestelmän puolelta epäonnistui, kun kirjasin oli upotettuna dokumenttiin. Todennäköisesti InDesign oli koodannut monenkirjavan joukon kirjasimia tavalla, joka ainakin hämmensi kolmannen osapuolen ohjelmaa, ellei niitä ollut koodattu peräti väärin (Preflight ei tosin löytänyt virheitä).

Palasimme siis askeleen taaksepäin alkuperäiseen PDF-tiedostoon, josta poistimme upotetun Minion Pro -kirjasimen edellä kuvaillulla Adobe Acrobatin optimointitoiminnolla. Upotetun kirjasimen poistamisen jälkeen PDF Architect suostui muokkaamaan tiedostoa oikealla kirjasimella:

* Päivämäärät saattavat vielä muuttua.

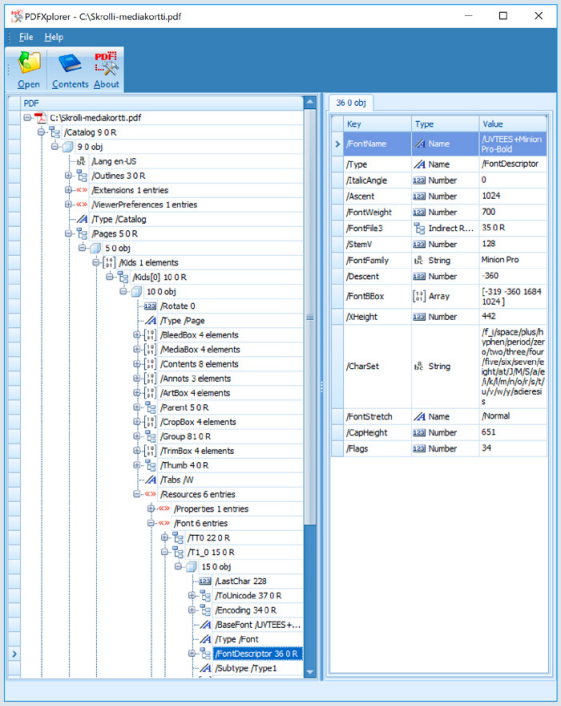
Lisätietoja

Mediamyynti: **Toni Kuokkanen**
puh: **+358 40**
email: **myynti@skrolli.fi**

Harmaita hiuksia aiheuttavia ongelma syntyi tosin tälläkin tekniikalla, nimittäin hieman alkuperäisestä poikkeava kirjasimen paksuus. Tämä on seurausta siitä, että on muokkausohjelman vastuulla tallentaa PDF-dokumenttiin kirjasinten asetukset, välitykset ja mittasuhteet. PDF Architect tekee tämän hieman eri tavalla kuin alkuperäisen PDF-dokumentin luonut InDesign.

Paluu työkalupakille

Lopulta tapauksen parhaaksi ratkaisuksi osoittautui useamman PDF-muokkaimen kokeileminen. Infix Pron käsinmuokkaustoiminto ei koskenut sen enempää esimerkkitiedoston rivitykseen kuin sotkenut kirjasinta tai sen välitystäkään – myös upotetut kirjasimet saivat jäädä paikoilleen. Kuten edellä selviää, Infix Prokin hajottaa dokumentin joissain tapauksissa, mutta tällä kertaa lopputulos oli täydellinen. Eron InDesign-versioon huomaa vain tiedostokoosta, sillä Infix Pro koodaa PDF:n hieman eri tavalla.



Skrollin mediakortin PDF-rakennetta avattuna ilmaisessa PDFXplorer-ohjelmassa.

* Päivämäärät saattavat vielä muuttua.

Lisätietoja

Mediamyynti: **Toni Kuokkanen**
puh: **+358 40**
email: **myynti@skrolli.fi**

Infix Pro suostui muuten kirjasinta vaihtamatta muokkaamaan myös tekstiä, jonka kirjasin oli upotettu dokumenttiin, mutta jonka fonttia edes Acrobat ei löytänyt koneelta. Fontti oli kyllä asennettu, mutta sen nimi oli ilmeisesti koodattu PDF-tiedostoon hieman väärin. Ohjelman valinnalla voi siis olla paljon väliä, sillä muut edellä mainitut ohjelmat vaihtoivat tällaisessa tilanteessa muokkaukseen käytettyä kirjasinta väkisin, mikä tietysti pilaa diskreetin muokkausyrityksen.

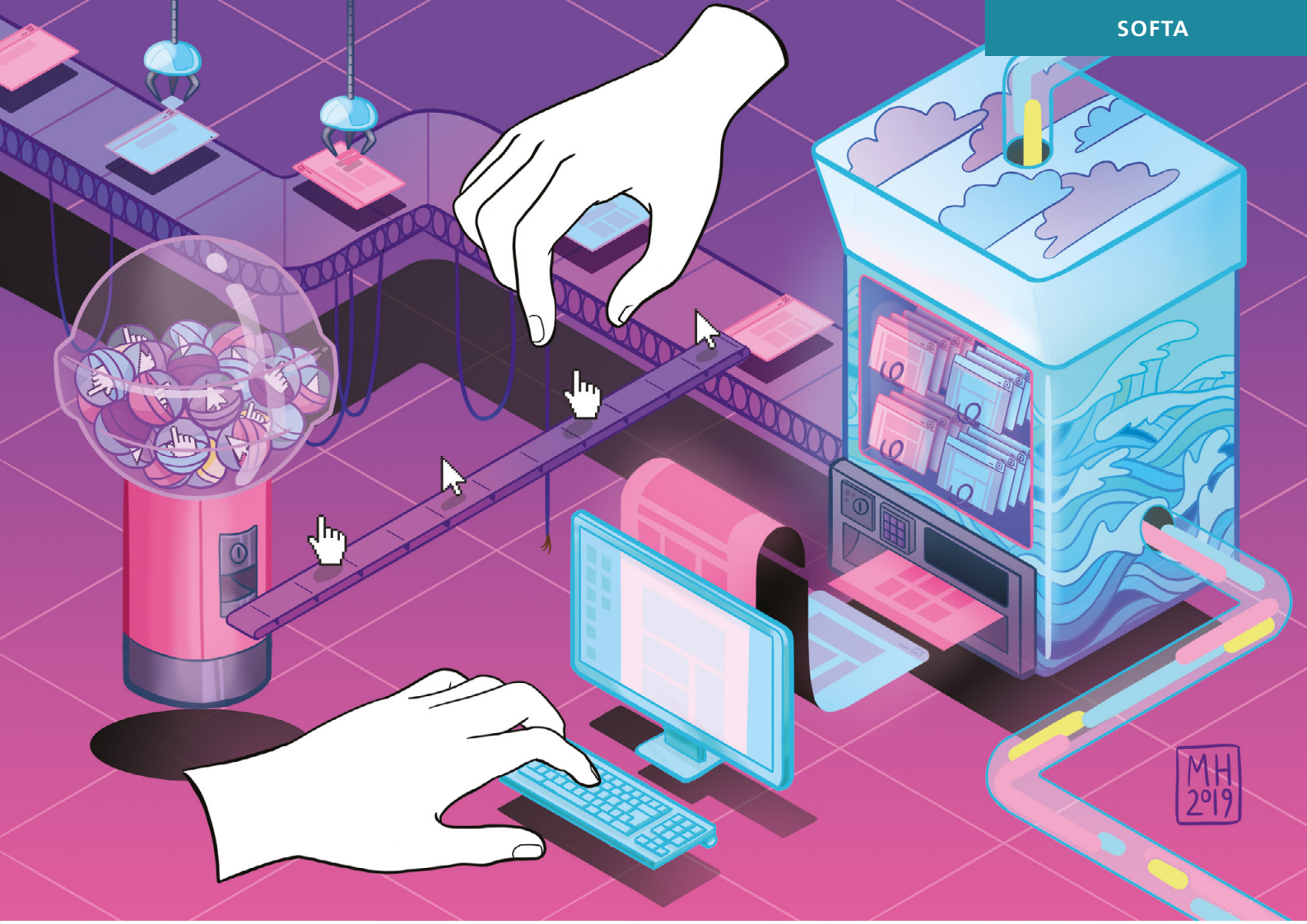
Empiirisen kokemuksen perusteella PDF:n muokattavuus jälkikäteen riippuukin muokkausohjelman lisäksi suuresti siitä, miten tiedosto on koodattu. Jopa samalla ohjelmalla luoduissa tiedostoissa muokkaus voi toimia yhdessä kohtaa hyvin ja epäonnistua toisessa täysin. Tässä tapauksessa esimerkkidokumentin jatkomuokkaus sujui kaikissa mainituissa ohjelmissa hyvin, kun tekstiä oli ensin kerran muokattu ja tiedosto tallennettu eli koodattu uudelleen Infix Prossa.

Voisi kuvitella, että Adoben oma InDesign ja Acrobat pelaisivat paremmin yhteen, mutta PDF-tiedostojen muokkauksen maailmassa tällaisista harhaluuloista kannattaa luopua. Voi kuitenkin lohduttautua sillä, että kun saa jollain kikkasarjalla PDF-dokumentin muokattua (paitsi jos todella rikkoo sen käsin), tämän jälkeen käytännössä kaikki ajan tasalla olevat PDF-lukijat kyllä toistavat sen identtisenä.

Koodaa itse!

Adobe vapautti PDF-spesifikaation vuonna 2007 International Organization for Standardization -järjestön (ISO) hallinnoitavaksi, eikä sen käytöstä tarvitse enää maksaa rojalteja. Viimeiseksi Adoben hallinnoimaksi versioksi jäi PDF 1.7 ja nykyisin Adobe on vain yksi ISO:n komiteajäsenistä. Standardointiprosessi oli pitkä, joten käytännössä PDF 1.7 on edelleen de facto -standardin asemassa. Löydät sen yksityiskohtaisen kuvauksen täältä: adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf

PDF 2.0 -standardi (ISO 32000-2:2017), aikaisemmat ISO:n alaisuudessa julkaistut osastandardit (kuten PDF/A, PDF/E ja PDF/UA) sekä skannauskäyttöön suunnattu PDF/raster löytyvät esimerkiksi PDF Associationin kautta: pdfa.org/resource



Vaihtoehto Wordpressille

Staattisen HTML:n paluu

Kuin varkain, vanha kunnon HTML-sivu on tehnyt paluun. Älä silti vielä kaiva under construction -giffejä perintö-cd-romilta. Nämä eivät ole vanhempiesi webbisivuja.

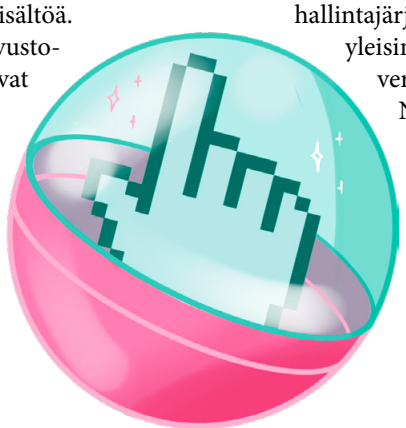
Teksti: Mika Haulo
Kuvat: Mikael Heikkanen, Janne Sirén, Mika Haulo

Erilaisista sisällönhallintajärjestelmistä alkoi tulla suosittuja keinoja rakentaa verkkosivustoja 2000-luvun alun jälkeen webbiteknologioiden kehittymisen myötä. Kymmenisen vuotta myöhemmin tietokantavetoisten CMS-järjestelmien rinnalle alkoi ilmestyä staattisia sivustogeneraattoreita vaihtoehtoiseksi tavaksi hallita nettisivujen sisältöä.

Mutta mitä staattiset sivustogeneraattorit oikeastaan ovat ja miten ne toimivat?

Ennen kaikki oli staattisemmin

Alunperin kaikki verkkosivut olivat staattisia HTML-sivuja. Siksi niitä kutsuttiin vain verkkosivuiksi. Myöhemmin, kun skriptikielet, tietokannat ja muut webbiteknologiat tulivat käyttöön, alettiin puhua dynaamisista verkkosivuista. Lisämääreellä dynaaminen painotettiin, että sivun sisältö ei määräytynytkään suoraan sen kirjoitushetkellä, vaan silloin kun käyttäjä latasi sen selaimeensa.

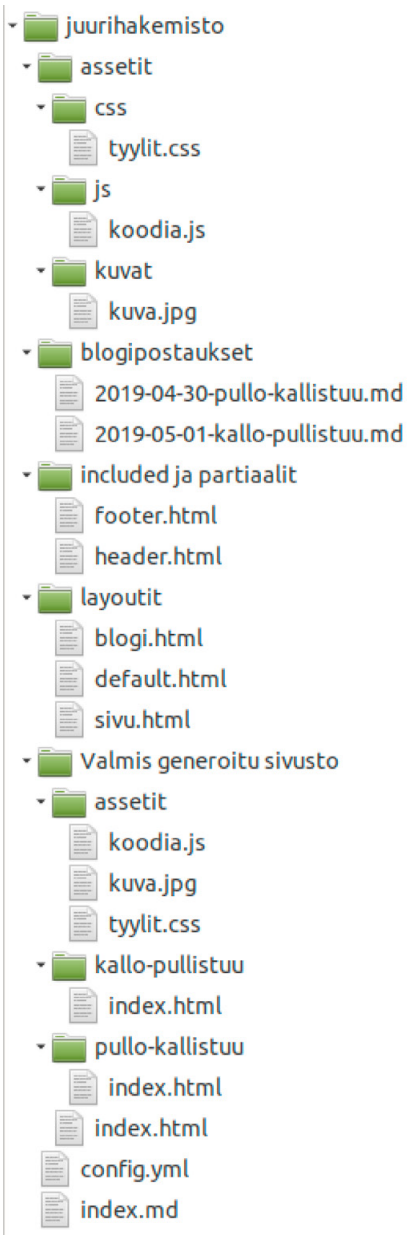


Ajan saatossa erilaisista sisällönhallintajärjestelmistä tuli yleisin tapa päivittää verkkosivustoja. Niiden käyttö kun ei vaatinut taitoa käsitellä HTML-dokumentteja suoraan. Sisältöasioiden hoitamiseen riitti sisällöntuotannon osaaminen, ja toteutus jäi teknisiin asioihin keskittyvien koodareiden työstettäväksi.

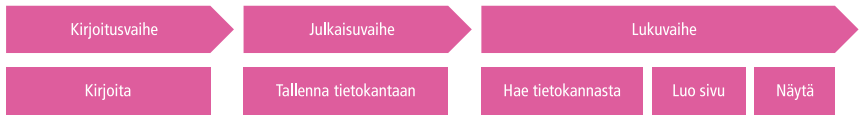
Sisällönhallintajärjestelmät ovat tehneet verkkosivujen sisällön muokkaamisen helpoksi, mutta toisaalta samaan aikaan vaikeuttaneet ylläpitoa ja hostausta asettaen näille suurempia vaatimuksia perinteisiin staattisiin sivuihin verrattuna. Esimerkiksi tietoturvan ja suorituskyvyn seuranta ovat nousseet aktiivisten töiden listalle, koska päivittämätön crm-ohjelmisto saattaa johtaa sivuston ”korkkaamiseen” ja tökkivä tietokantamoottori saattaa karkottaa kärsimättömän surffailijan pois ennen sivun latautumista – tai pahimmillaan estää sivun latautumisen kokonaan.

Takaisin (jalostetuille) juurille

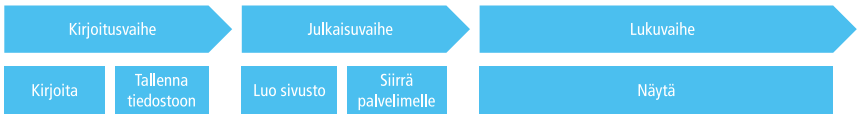
2010-luvun alkupuolella tietokantavetoisten sisällönhallintajärjestelmien rinnalle alkoi ilmestyä kevyempiä vaihtoehtoja eli niin sanottuja staattisia sivustogeneraattoreita. Toinen yleinen englanninkielinen termi näille järjestelmille on *flat file CMS*, joka viittaa siihen, että kaikki tarvittava tieto kirjoitetaan nimenomaan tiedostoihin eikä tietokantaan.



Tyypillinen sivustoprojektin hakemistorakenne.



Kirjoitus-, julkaisu- ja lukemisvaiheiden keskinäinen suhde tietokantapohjaisissa julkaisujärjestelmissä.



Kirjoitus-, julkaisu- ja lukemisvaiheiden keskinäinen suhde staattisissa sivustogeneraattoreissa.

Jälleen kerran lisämääre staattinen on mukana painottamassa eroa nykyiseen de factoon: lopputuloksena on dynaamisen verkkosivuston sijaan yksi tai useampia perinteisiä HTML-dokumentteja.

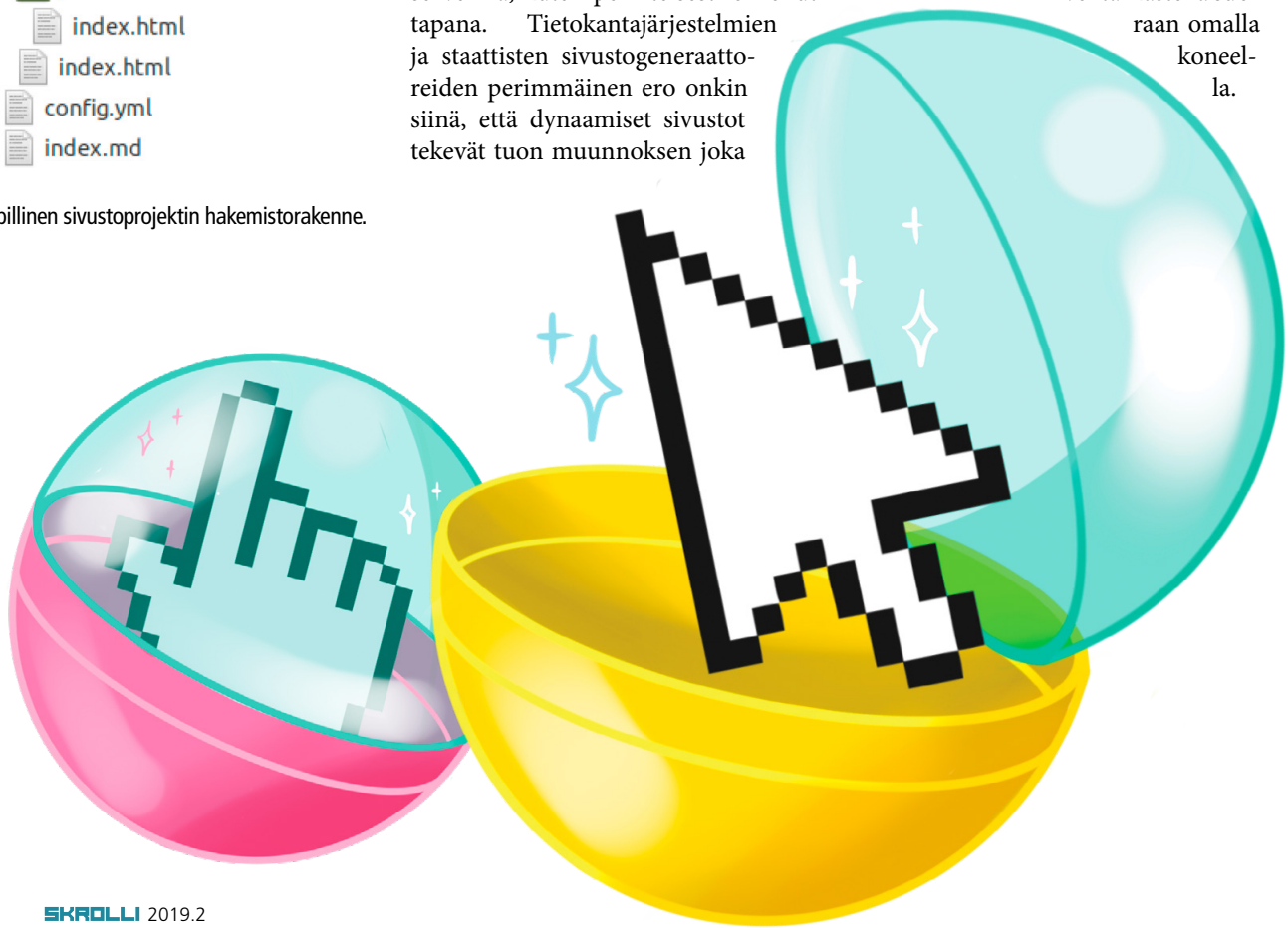
Staattisten sivustogeneraattoreiden toimintaperiaate on erottaa toisistaan toiminnallinen osa, jota tarvitaan eniten sisällön muokkaamisessa, ja näyttöosa, johon usein riittää yksinkertaisempi toteutus – loppujen lopuksi jokainen verkkosivu on selaimelle saapuessaan käännetty staattiseksi HTML-koodiksi.

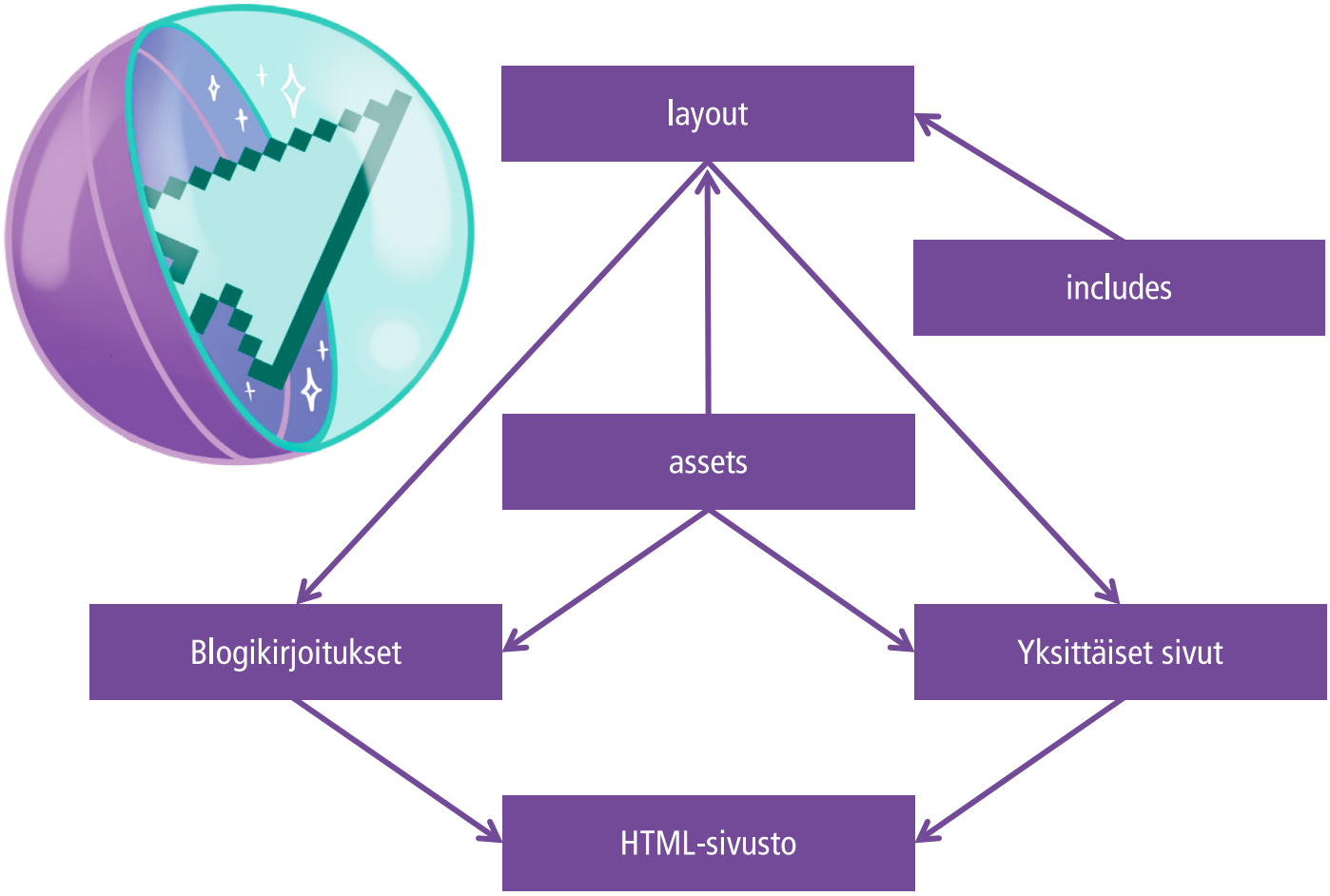
Myös Javascript-teknologioiden kehittyminen on mahdollistanut sen, että monet dynaamiset elementit voidaan hoitaa suoraan selaimessa, eikä serverillä, kuten perinteisesti on ollut tapana. Tietokantajärjestelmien ja staattisten sivustogeneraattoreiden perimmäinen ero onkin siinä, että dynaamiset sivustot tekevät tuon muunnoksen joka latauskerralla erikseen, kun taas staattiset generaattorit tekevät sen vain silloin kun sisältöä muutetaan.

Miten kaikki toimii?

Staattiset sivustogeneraattorit perustuvat usein Markdown-muotoilukielen hyödyntämiseen. Puhtaan HTML-kielen sijaan otsikot, boldaukset, linkit, kuvat ynnä muut lisätään dokumenttiin yksinkertaisemmalla Markdown-muotoilulla. Toisaalta Markdown-dokumenteissa usein sallitaan myös suoraan HTML:n käyttö, joten monimutkaisempiakin rakenteita voi tehdä, jos Markdown itsessään ei niihin taivu.

Kehitysvaihetta varten sivustogeneraattoreissa voi olla sisäinen palvelin, jonka avulla tekeillä olevaa sivustoa voi tarkastella suoraan omalla koneella.





Sivustogeneraattorin toimintaperiaate. Sisältö (esim. blogikirjoitukset) kirjoitetaan markdown-dokumentteihin, joiden metadata määrittää, mitä layoutia käytetään. Sekä dokumentti että sommittelu (layout) voivat sisältää kuvia. Sivustogeneraattori yhdistää näitä eri palasia toisiinsa tiedostojen sisältämien viittausten perusteella. Lopputuloksena on nippu HTML-tiedostoja.

Julkaisua varten sivustosta luodaan ns. julkaisuversio, joka sitten siirretään webbihotelliin tai muulle vastaavalle verkkopalvelimelle. Lopputuloksena on perinteisen näköinen ja oloinen kokoelma HTML-dokumentteja kuvineen päivineen, eikä lopputuloksesta voi välttämättä suoraan päätellä, mitä työkalua sivuston luomiseen on käytetty. Yhdestä näkökulmasta staattiset sivustogeneraattorit ovatkin eräänlainen HTML-muokkausohjelmisto, tosin luonteeltaan hyvin erilaisia kuin perinteiset HTML-muokkaimet (Dreamweaver jne.).

Sivustogeneraattorit myös pyrkivät poimimaan parhaita paloja dynaamisten sisällönhallintajärjestelmien luomista käytännöistä. Hyvä esimerkki on sivuston sommittelu (*layout*). Sen sijaan että jokaiselle sivulle tehtäisiin ulkoasun muotoilu erikseen, erottaa sivustogeneraattori sisällön ja muotoilun selkeästi toisistaan. Kääntövaiheessa sisältödokumentit ja layout-dokumentit vain yhdistetään yhdeksi kokonaisuudeksi.

KATSAUS VALIKOIMAAN

Tätä nykyä erilaisia sivustogeneraattoreita on kymmeniä. Niillä kaikilla on samankaltainen perusperiaate, ja erot syntyvät esimerkiksi käytetystä ohjelmointikielestä. Tunnetuimpia sivustogeneraattoreita ovat Jekyll ja Pico. Jekyll-vetoisia sivuja moni koodaaja lukee aivan huomaamattaan, sillä Githubin sivut (Github Pages) perustuu Jekylliin.

Tavallisten verkkosivustojen lisäksi staattisia sivustogeneraattoreita hyödynnetään myös esimerkiksi dokumentointisivustoissa. Dokumenttisivustoihin erikoistuneita generaattoreita ovat esimerkiksi MkDocs ja Pelican.

Nimi	Toteutuskieli	Template-kieli	Lisenssi
Jekyll	Ruby	Liquid	MIT
Pico	PHP	Twig	MIT
Hugo	Go	Go	Apache 2.0
Next	Javascript	React	MIT
Middleman	Ruby	Erb, Tilt, Haml	MIT
Gatsby	Javascript	React	MIT
Mkdocs	Python	Jinja 2	BSD 3-clause
Pelican	Python	Jinja 2	AGPL 3.0

Kattava lista eri vaihtoehdoista: www.staticgen.com

SIETÄMÄTTÖMÄT INDEKSIT

Miksi ohjelmointi on nykyään niin hankalaa?

Teksti: Antti Laaksonen
Kuva: Jakke

Opetan yliopistolla viikoittain ohjelmointia lapsille, ja työni on oikein antoisaa yhtä asiaa lukuun ottamatta: ei ole mitään ohjelmointikieltä, joka sopisi hyvin ohjelmoinnin opettamiseen. Toivelistallani on vain kaksi kohtaa: kielessä tulisi olla helposti ymmärrettävä syntaksi, ja lisäksi kielen tulisi olla yleisesti käytetty. Ikävä kyllä mikään nykyinen kieli ei täytä näitä vaatimuksia.

Yksi ensimmäisistä harjoituksistamme on tulostaa luvut 1–10. Tässä tuleekin vastaan ensimmäinen ongelma: for-rakenteen hankaluus. Esimerkiksi Pythonissa tarvittava koodi on seuraavanlainen:

```
for i in range(1,11):
    print(i)
```

Haluamme tulostaa luvut 1–10, joten laitamme rajaksi 1–11. Loogista, eikö vain? Itse asiassa en ole vielä tavannut yhtään aloittavaa ohjelmoijaa, joka olisi pitänyt tätä loogisena. Tässä näkyy yksi nykyisten kielten hankaluuksista: puoliavoimet välit. Koska välin viimeinen luku ei oikeasti kuulu siihen, jokainen väli tulee ilmoittaa yhtä liian suurena.

Myöhemmin opimme taulukoiden käsittelyä, jolloin vastaan tulee uusi kummallisuus. Kun otamme taulukosta kohdassa 3 olevan luvun, emme saa kolmatta lukua vaan neljännen luvun. Tässä on kyse siitä, että taulukoissa ja muissa rakenteissa indeksointi alkaa nollasta. Nyt siis vaihtelun vuoksi tulee antaa yhtä pienempi arvo, kuin mitä halutaan oikeasti.

Kokeneelle ohjelmoijalle yllä olevassa ei toki ole mitään hankalaa, mutta tällaiset asiat vaikeuttavat oppimista. Ylättävää kyllä, vielä 1980-luvulla mitään tällaisia ongelmia ei ollut, vaan vallitsi todellinen ohjelmoinnin opettajan paratiisi. Silloin välit olivat suljettuja ja indeksointi järkevää. Esimerkiksi ennen vanhaan suositussa Pascal-kielessä luvut 1–10 tulostetaan näin:

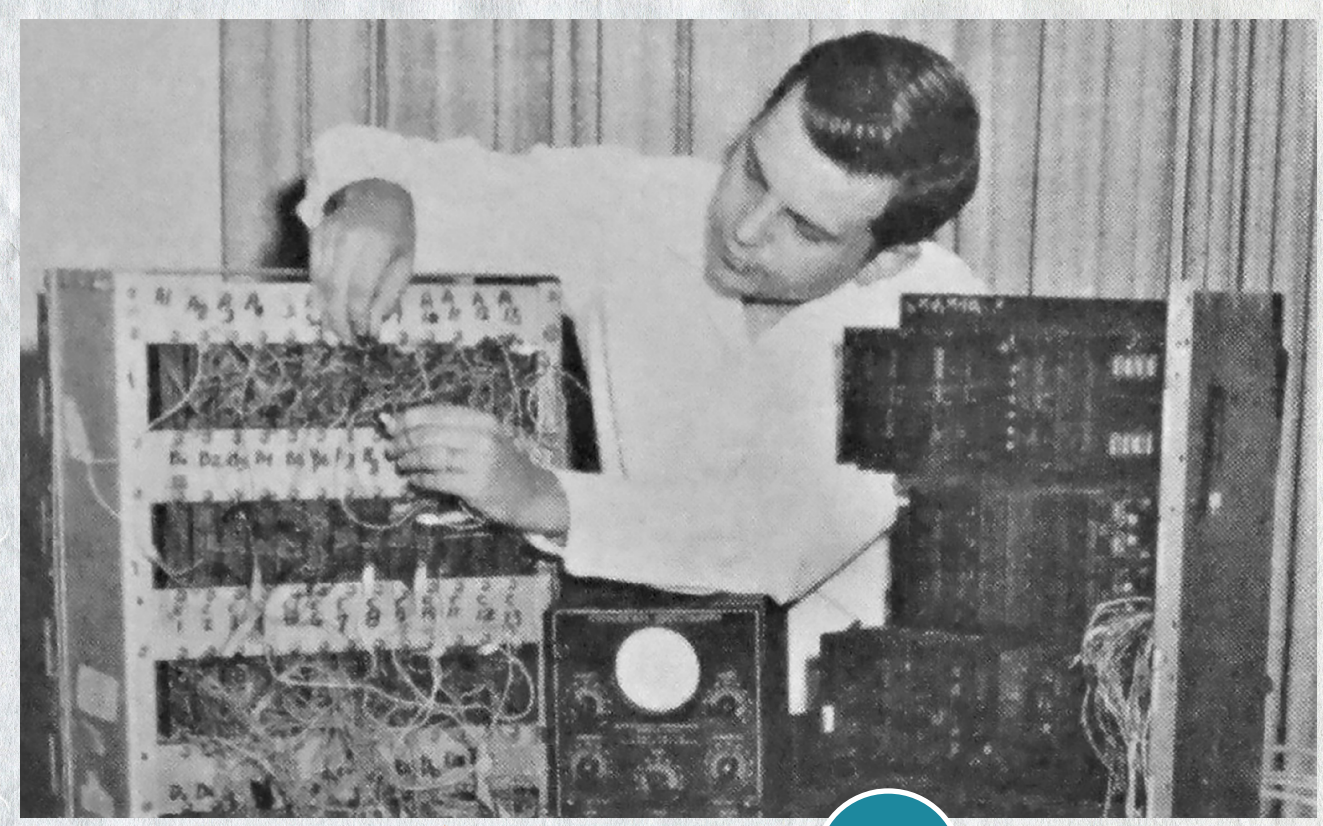
```
for i := 1 to 10 do
    writeln(i);
```

Eli kun haluamme tulostaa luvut 1–10, annamme väliksi 1–10. Yksinkertaista mutta nerokasta. Entä taulukon indeksointi? Nyt tulee todellinen jymyjuttu: Pascalissa saamme päättää itse, mikä on taulukon ala- ja yläraja. Esimerkiksi jos tarvitsemme taulukon, jossa pienin indeksi on 3 ja suurin indeksi on 8, voimme luoda juuri tällaisen taulukon. Kiista oikeasta alkuindeksistä tyrehtyy siihen, että jokainen voi indeksoida haluamallaan tavalla.

Nämä eivät ole vain Pascal-kielen ominaisuuksia, vaan samat periaatteet olivat muissakin suosituissa kielissä 1980-luvulla ja aiemmin. Vaikuttaa siis siltä, että ohjelmointikielet ovat menossa huonoon suuntaan: ennen kielet olivat ymmärrettäviä, nykyään ne ovat hankalia. Miten tällainen kehitys voi olla mahdollinen?

En ole suinkaan ainoa, joka on ihmetellyt asiaa, ja jos etsii tietoa netistä, alkaa löytyä selityksiä muotoa "ohjelmoijat ajattelevat näin" ja "meitä viisaammat ihmiset ovat päättäneet näin". Ehdottomasti viitatuin viisaampi päättäjä on **E. W. Dijkstra**, joka kirjoitti vuonna 1982 tekstin "Why numbering should start at zero". Dijkstran mukaan nolla on luonnollisin luku, joten indeksointi tulee aloittaa siitä, ja välien puolestaan tulee olla puoliavoimia, koska silloin välin alkioiden määrä on sama kuin loppu- ja alkuindeksin erotus.

Dijkstran unelma alkoi toteutua pikkuhiljaa, ja nykyään puoliavoimet välit ja nollaindeksointi vallitsevat lähes kaikissa kielissä. Mutta jokainen ohjelmointia opettava tietää, ettei tämä ole oikeasti toimiva systeemi, vaikka niin halutaan väittää. Toki kaikkeen tottuu, mutta ohjelmoinnin oppiminen on muutenkin vaikeaa ja paljon turhaa lisävaikeutta tulee kielen hankalasta syntaksista. Mutta mistä löytyisi nykyajan Dijkstra, joka alkaisi johtaa kehitystä toiseen suuntaan?



OSA II

Sarja jatkuu

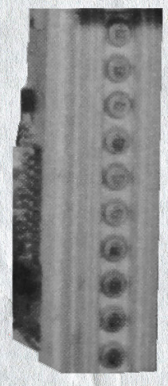
Kuinka rakensin tietokoneen 1960-luvulla

Edellisessä Skrollissa alkaneessa juttusarjassa ohjelmistosuunnittelija Joseph Watson kertoo, millaista oli rakentaa oma tietokone aikana ennen mikroprosessoria.

Teksti ja kuvat: Joseph Watson
Käännös: Mikko Heinonen

Viime kerralla tarina jäi siihen, että olin hankkinut **Charles ”Bud” Cokerin** ylijäämämyymälästä kasan piirilevymoduuleita tietokoneeni rakennuspalikoiksi.

Moduulit sisälsivät kiikkuja, portteja, monostabiileja multivibraattoreita ja muutamia muita elementtejä. Tähän aikaan piistä tehdyt transistorit ja diodit olivat vielä harvinaisia. Kaikki moduuleilla olevat transistorit ja diodit oli tehty germaniumista, ei piistä. VOM-yleismittarin (voltti-ohmi-milliampeeri) avulla selvitin pian, että moduuleissa lähes yksinomaisesti käytetyt 2N414-transistorit olivat PNP-tyyppisiä. Ne tarvitsivat negatiivisen syöttöjännitteen. Veikkasin, että –12 volttia voisi olla sopiva syöttöjännite, ja etsin sopivaa virtalähdettä ylijäämämyymälöistä. Bud pelasti jälleen päivän ja myi minulle reguloidun 12 voltin virtalähteen, joka tuotti 20 ampeerin virran.

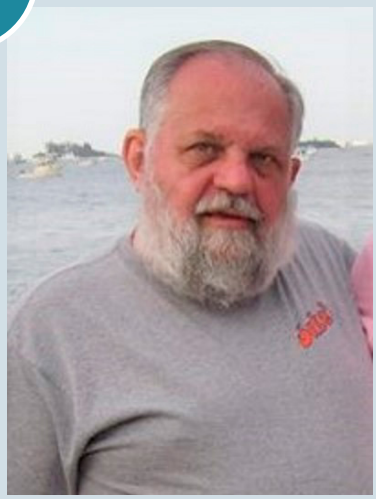


12 voltin, 20 ampeerin reguloitu virtalähde.

Minulla on edelleen tallessa tämän vanhan projektin moduuleita. Valmistellessani tämän tarinan kirjoittamista etsin niitä ja löysin ne samasta kasasta monien muiden vanhojen osien kanssa.

Piirimoduulit rakennuspalikoina

Piirilevyt ovat useimpiin nykyaikaisiin tuotteisiin verrattuna melko karuja. Niiden pohjamateriaali on fenoli (eikä laminoitu lasikuitu ja epoksi), komponentit on selvästi ladottu käsin (eikä tietokoneohjelmalla), levyt ovat yksipuolisia, eikä niissä ole juotteenes-



Joseph Watson liittyi Skrolli-kirjoittajien joukkoon kansainvälisen *Skrolli International Edition* -lehtiprojektimme yhteydessä. Tarina hänen myöhemmästä mikroprosessoria käyttäneestä mikrotietokoneestaan on julkaistu Skrollin numerossa 2016.4 (pdf-lehti: skrolli.fi/numerot).

topinnoitetta tai silkkipainatusta, josta komponentit voisi tunnistaa. Johtimet ovat 1,5 mm:n levyisiä ja ne on päällystetty koko pituudeltaan lyijyä sisältävällä tinalla.

En ollut aiemmin tiennyt, mihin tarkoitukseen nämä moduulit oli alkujaan rakennettu. Selvää oli, että ne sopivat moniin erilaisiin logiikkapiirejä käyttäviin projekteihin.

Internetin avulla sain hiljattain selvitettyä moduulien alkuperän. Ne oli suunniteltu Yhdysvaltain standardoimisviraston käyttöön eri hankkeita varten. Yksi hankkeista oli Ceilometer-nimellä tunnettu laite. Se käytti pyörivää, vilkkuvaa valokeilaa, joka suunnattiin ylös pilviä kohti. Pilvistä heijastuva valo tunnistettiin ja analysoitiin, minkä perusteella voitiin määrittää pilvien korkeus. Ceilometer oli tietokone, jota käytettiin vain yhteen tarkoitukseen. Se oli johdotettu suorittamaan vain yhtä tehtävää, eikä se ajanut mitään ohjelmaa.

Minulle selvisi hiljattain myös, että −12 volttia oli moduulien suunnittelujännite. Olin arvannut oikein.

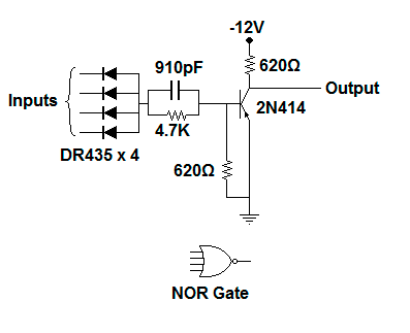
50-vuotiaita logiikkapiirejä

Kiinnostuneille kerrottakoon, että moduulien yleisimmät piirityypit ovat NOR-portti (eli käänteinen OR-portti) ja kiikku (eli bistabiili multivibraattori).

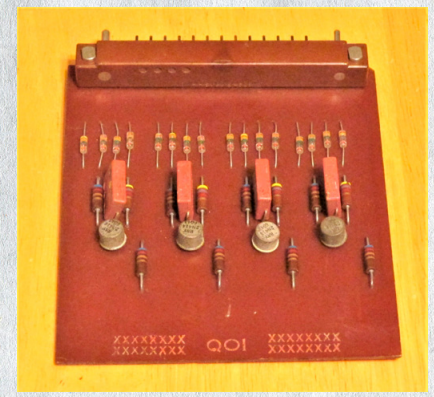
Tässä on NOR-portin kytkentä (muista, että tässä järjestelmässä logiikan ”1” on −12 volttia ja logiikan ”0” on 0 volttia).

Jos 1 tai useampia tuloja on −12 V (logiikan ”1”), lähtö on 0 V (logiikan ”0”). Jos kaikki tulot ovat 0 V (logiikan ”0”), lähtö on −12 V (logiikan ”1”).

Jokainen neljän NOR-portin moduuli sisältää 4 erillistä 4 tulon NOR-porttia.

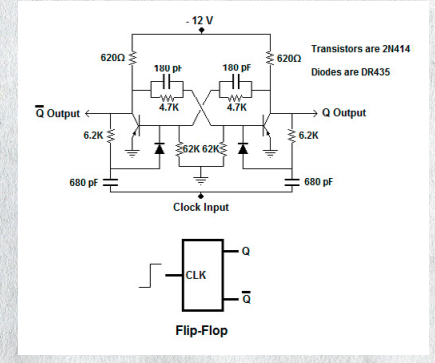


Piirikaavio 4 tulon NOR-portille.



Neljän NOR-portin moduuli.

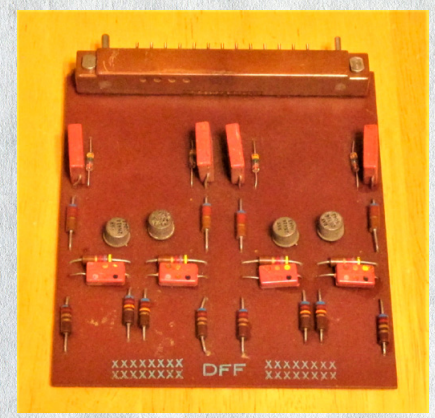
Moduuleilla oleva perusmallin kiikkukytkentä on toteutettu niin, että kytkemällä sen liittimen nastoja eri tavoin sen saa toimimaan RS-kiikkuna, binäärilaskurina, siirtorekisterinä tai minä tahansa näiden yhdistelmänä. Esimerkiksi binäärilaskuriksi määritettynä kiikkupiiri näyttäisi seuraavalta. Tämä kiikku vaihtaa kahden tilansa välillä aina kun sen kellotulon jännite vaihtuu positiiviseen suuntaan, kuten esim. −12 voltista (logikaa ”1”) 0 volttiin (logiikan ”0”).



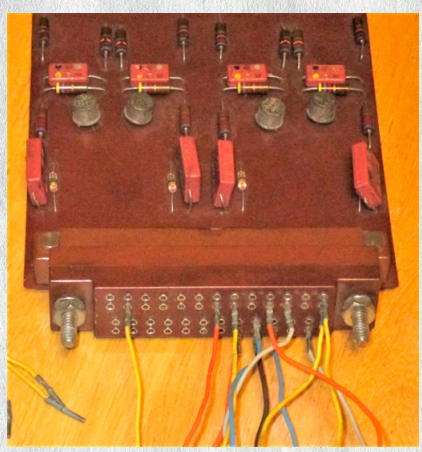
Piirikaavio binäärilaskurin kiikulle.

Kiikkuja on yleensä kaksi yhdellä logiikkamoduulilla.

Juttua kirjoittaessani aloin miettiä, toimisiko joku näistä moduuleista vie-



Kahden kiikun moduuli.



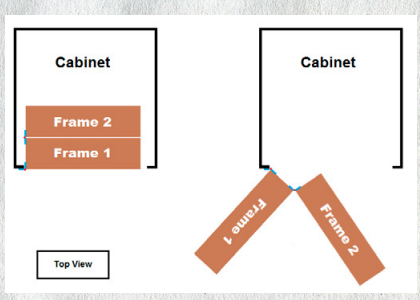
Moduuli, jossa on yksi kiikku kytkettynä testausta varten.

lä. Valitsin kiikkupiirin, kytkin siihen riittävästi johtoja, laitoin virrat päälle ja kokeilin. Moduuli toimi aivan kuin 50 vuotta sitten.

Tein moneen logiikkamoduuliin pieniä muutoksia: poistin tai lisäsin komponentteja, muutin vastusten arvoja tai jopa lisäsin portteja tai transistoreita.

Moduulikehikoiden asennus

Lisäsin jokaiseen moduuliliittimillä varustettuun kehikkoon vasempaan reunaan pitkän saranan. Kiinnitin ensimmäisen kehikon kaappiin saranan avulla ja sitten toisen kehikon ensimmäisen takaosaan. Näin molemmat kehikot sai tarvittaessa käännettyä ulos, jotta liitinten takana olevia johtimia sai muutettua.



Yksityiskohta kehikosta.

Järjestelmän kello

Kaikkia koneen toimia ohjasivat kellopulssit, jotka saatiin vapaasti toimivalta, astabiililta multivibraattorikellolta. Kello ei ollut kideohjattu, vaan sen nopeutta sai säädettyä potentiometrillä. Kääntämällä säädintä sain vähennettyä suoritustehoa merkittävästi. Lisäsin jopa kytkimen, jolla kellon sai pysäytettyä kokonaan. Jokainen kytkimen kääntäminen suoritti operaation seu-

raavan vaiheen. Sain valvottua ja tarkkailtua koneen jokaista toimintoa koneen valojen ja jännitemittarin avulla.

Nopeimmillaankin kellotaajuus oli todennäköisesti alle 100 kHz.

Sarjalaskentaa

Nykyisin kaikki tietokoneen ALUn (aritmetiikka- ja logiikkayksikkö) tekemät laskutoimitukset tehdään rinnakkaistamalla. Operaatio suoritetaan tavun tai sanan kaikille biteille yhtä aikaa.

Esimerkiksi kahden 8-bittisen luvun yhteenlaskussa käytetään kahdeksaa identtistä piiriä. Jokainen piiri laskee yhden bitin näistä arvoista ja tuottaa yhden bitin tuloksesta.

Jos kuitenkin halutaan vähentää prosessiin tarvittavien logiikkaelementtien määrää dramaattisesti, eräs tapa on tehdä operaatiot bitti kerrallaan. Esimerkiksi yhteenlaskussa voidaan käyttää yhtä piiriä, joka pystyy lisäämään yhden bitin 8-bittisestä arvosta, ja tehdä sitten operaatio 8 kertaa peräkkäin (kerran jokaista bittiä kohti). Tätä kutsutaan sarjalaskennaksi, ja se tietenkin vaikuttaa merkittävästi laskentanopeuteen.

Tämän sarjalaskentaperiaatteen mukaisesti tietokoneeni kolmessa tärkeimmässä rekisterissä (AC, MD ja PH) oli kaikissa 10-bittinen siirtorekisteri, jolloin ALUn tarvitsi käsitellä vain yhtä bittiä kerrallaan, mutta siltä kesti 10 kellojaksoa suorittaa yksinkertaisia operaatioita kuten LOAD, ADD tai SUB.

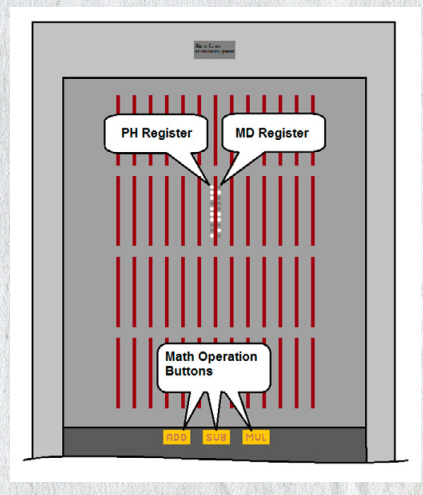
Valoja tarvittiin

Tarvitsin ehdottomasti jonkin keinon nähdä koneen rekisterien tilan. Kokosin siis hehkulampuista kolme 10-bittistä näyttöä. Muokkaamalla yhtä

Kotitekoinen 20 lampun näyttömoduuli.

Näyttömoduuli edestä katsottuna.

moduulikorttia sain aikaan ohjaimet ja lamput, jotka näyttivät MD- ja PH-rekisterin. Rekisterien lamput asensin piirilevyn etureunaan, jolloin ne näkyivät koneen etuosasta.



Kehikon 1 logiikkaosion luonnos, jossa lamput näkyvät.

Toinen 10 lampun sarja oli vaakatasossa alla kuvatun oktaalisyöttölaitteen alapuolella. Nämä lamput näyttivät joko AC:n tai PC:n tilan riippuen oktaalisyöttölaitteen kytkimen asennosta (katso alla olevaa oktaalisyöttölaitteen kuvaa).

Kirjoituskone

Tulostuslaitteeksi Bud myi minulle IBM:n Standard Electric Model B -kirjoituskoneen (jossa oli vasarat, ei golfpallon näköistä pyörivää kirjoituspäätä). Se oli valmistettu erityisesti jotakin tiettyä käyttötarkoitusta varten, sillä käytössä oli vain isot kirjaimet, eikä pieniä saanut käyttöön ilman laitteen modifiointia. Omaan käyttööni pelkät isot kirjaimet riittivät, varsinkin koska tarkoitukseni oli tulostaa vain oktaalilukuja 0–7 sekä välilyöntejä ja rivinvaihtoja.

Kirjoituskoneen vasarat ja värinauha.

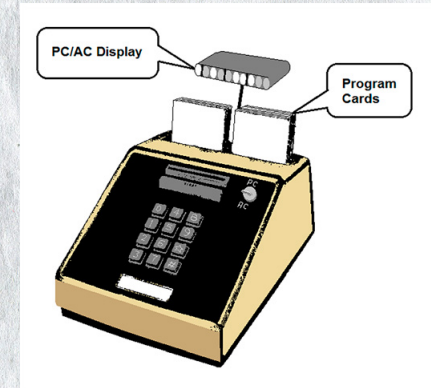
Asensin kirjoituskoneen kymmenen näppäimen alle pienet solenoidit, jotka vetivät niitä alaspäin kirjoitettaessa. Logiikkapiirit ohjasivat näitä solenoideja. Lisäsin myös rajakytkimen, joka tunnisti milloin vaunu läheni rivin loppua. Kun koneen ohjelmassa kohdattiin TYPE-komento, ohjelman suoritus pysähtyi muutamaksi sekunniksi ja järjestelmä kirjoitti AC:n sisällön 4-bittisenä oktaalilukuna ja sen perään joko välilyönnin tai rivinvaihdon riippuen siitä, oliko rajakytkin aktivoitunut vai ei. Sitten ohjelman suoritusta jatkettiin.

Järjestelmässä ei ollut I/O-portteja, ei ready-lippuja, ei keskeytyksiä, ei mitään ohjausrekistereitä eikä bittejä minkään kytkemiselle päälle tai pois. Se oli niin yksinkertainen kuin vain mahdollista.

Kirjoituskone oli hyllyllä, jonka sai vedettyä kaapista ulos kuulalaakeroitujen rullien päällä. Koska kirjoituskoneen vaunu liikkui puolelta toiselle käytön aikana, se tarvitsi vaakasuunnassa enemmän tilaa kuin kaapissa oli. Järjestelyni avulla kirjoituskoneen sai vedettyä ulos koneen etuosasta käytön ajaksi ja varastoitua takaisin kaapin sisään, kun virta oli katkaistuna.

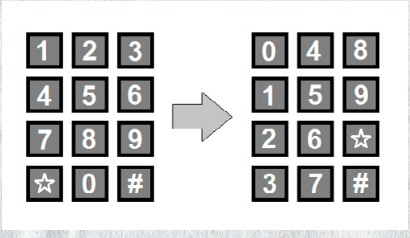
Oktaalisyöttölaite

Syöttölaitteeksi hankin vaihtokaupalla näppäinpuhelimen, jossa oli muovisia reikäkortteja käyttävä automaattinen numeron valinta. Tämä oli ensimmäinen koskaan omistamani äänitaajuuspuhelin, mutta en tarvinnut ääniä mihinkään, joten irrotin ääntä tuottavat virtapiirit.



Luonnos oktaalisyöttölaitteesta näyttöineen.

Näppäimistössä oli 12 painiketta järjestettynä samoin kuin nykyisissäkin puhelimissa. Kun jotakin niistä painettiin, mekanismissa sulkeutui kaksi kytkintä: yksi ilmaisi painetun painikkeen rivin, toinen sarakkeen. Päätin, että painikkeet olisi helpompi järjestää oktaalisyötteen vaatimaan kahdeksaan 3 bitin kuvioon, joten purin näppäimistön ja kokosin sen uudelleen seuraavasti:



Syöttölaitteen uusi järjestys oktaalikoodausta varten.

Tämän jälkeen tarvittiin vain kaksi diodia koodaamaan kaikki luvut (0–7) kolmella bitillä.

Oktaalisyöttölaitetta käytettiin kahteen tarkoitukseen: ohjelmien syöttämiseen sekä tietojen syöttöön käynnissä olevalle ohjelmalle.

Kun ohjelman suorituksen aikana kohdattiin INPUT-käsky, ohjelma pysähtyi ja odotti, että käyttäjä syöttää painikkeilla oktaaliluvun (enintään 4 numeroa). Syötettäessä luvut tallennettiin AC:hen, jonka tilaa saattoi seurata 10-bittiseltä näytöltä. Näppäimistön oikealla puolella olevat painikkeet (8, 9, * tai #) toimivat terminaattoreina, joilla ohjelman suoritusta sai jatkettua.

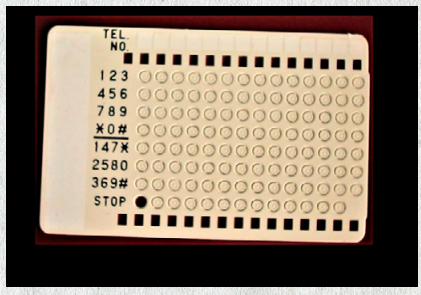
Kirjoituskoneen alapuolella olevan Reset-nappulan painaminen keskeytti käynnissä olevan ohjelman, tyhjensi 7-bittisen PC-rekisterin ja luki muistiosoitteen 0 osoitteen AC:hen, jolloin se näkyi oktaalisyöttölaitteeseen yhdistetyssä näytössä. Jos osoitteessa olevaa arvoa oli tarpeen muuttaa, näppäimistöltä manuaalisesti syötetty arvo korvasi AC:ssä olevan arvon. Kun terminaattorinäppäintä (8, 9, *, #) painettiin, AC tallennettiin muistiin käyttäen PC:ssä olevaa osoitetta. Sen jälkeen PC:n arvoa lisättiin ja seuraava muistilohko luettiin AC:hen. Tämän mekanismin avulla ohjelmaa voitiin manuaalisesti tarkastella/muokata/ladata. Kaikki ohjelmat alkoivat muistiosoitteesta 0.

Jos haluttiin nähdä PC:n arvo joko ohjelman suorituksen aikana tai ladattaessa uutta ohjelmaa, näppäimistön lähellä olevaa kytkintä kääntämällä 10-bittinen näyttö ilmaisi AC:n sijaan 7-bittisen PC:n sisällön.

Valitettavasti tietyn muistilohkon sisällön tarkasteluun ei ollut muuta tapaa kuin aloittaa osoitteesta 0 ja edetä yksi osoite kerrallaan, kunnes oltiin halutussa osoitteessa. Toisaalta koneen muistissa oli vain 128 osoitetta, joten rajoitus ei ollut merkittävä.

Oktaalisyöttölaite oli yhdistetty koneeseen kaapelilla, mutta se oli muuten varsin kannettava ja pidin sitä yleensä käsissäni koneen käytön aikana.

Automaattinen numeronvalitsin käytti erityisiä muovikortteja.



Muovinen automaattivalitsimen kortti.

Tekemällä reikiä sopiviin riveihin ja sarakkeisiin kortin sai koodattua soittamaan tiettyyn numeroon tai minun tapauksessani syöttämään oktaalilukujen sarjan.

Korttia käytettiin syöttämällä se näppäimistön yläpuolella olevaan koloon, työntämällä se alas asti ja painamalla Start-painiketta, jolloin sisäinen mekanismi veti kortin hitaasti takaisin ylös sulkien samalla rivien ja sarakkeiden kohdat kortissa olevien reikien mukaisesti.

Muoviset automaattivalintakortit vaikuttivat ideana mahtavalta, mutta niistä ei lopulta ollut minulle kovastikaan hyötyä. En muista koskaan tallentaneeni niille oikeaa ohjelmaa. Minulla on vieläkin pari pientä laatikollista käyttämättömiä kortteja.

Seuraavassa numerossa kerron, miten kone saatiin laskemaan ja muistamaan asioita.

Tarina jatkuu Skrollissa 2019.3.

Kuva tulostimena käytetystä IBM Standard Electric -kirjoituskoneesta.

Bitcoin tositilanteessa

Kuinka hyvin Bitcoin toimii käytännössä vuonna 2019? Testasin Bitcoinia oikeassa käyttötapauksessa.

Jarno Niklas Alanko

Sähköpostissani on hälyttävä viesti: verkkotunnukseni on vanhenemassa kahden viikon päästä, koska automaattinen uusiminen ei ole onnistunut luottokorttiongelman vuoksi. Verkkotunnukseni tarjoaja Namecheap ei osaa auttaa asiassa, vaan he kehottavat ottamaan yhteyttä pankkiin. Kuulostaa kovin vaivalloiselta.

Onneksi huomasin, että Namecheap tukee myös Bitcoin-siirtoja. Tämä tuntui helpoimmalta tavalta ratkaista ongelma. Samalla pääsin testaamaan, millaista Bitcoinilla maksaminen on nykyään käytännössä.

Lompakosta seuraavaan

Tehtävänä oli siis siirtää 25 dollaria omalta pankkitililtäni Namecheapille Bitcoin-verkon yli. Koska välityskuluihin menee kuitenkin rahaa, ostin kolikoita reilusti 50 eurolla. Hankin kolikot Coinbase-palvelusta, koska minulla oli sinne tunnukset jo valmiina. Osto sujui helposti luottokortilla, mutta Coinbase otti 2,31 euroa välistä.

Coinbasen käyttöliittymä tuntuu olevan suunniteltu ensisijaisesti sijoittajille. Erilaisia kolikoita on helppo ostaa ja myydä, ja portfolion arvon kehitykselle on hienot kuvaajat ohjauspaneelissa etusivulla, mutta kolikoiden siirtämisen mahdollistavaa alasivua sai hetken etsiä.

Oikean sivun löydettyäni kuitenkin valkeni, että kolikoita ei voi tuosta vaan siirtää Namecheapille, koska Namecheapin Bitcoin-tuki on toteutettu Bitpay-nimisen palvelun kautta. Coinbasessa ei ole suoraa tukea Bitpaylle, joten minun oli luotava Bitpay-lompakko ja siirrettävä kolikot sen kautta.

Lataan Bitpay-asiakasohjelman, joka löytyy Ubuntun Software Centeristä. Luon lompakon, johon liittyy yksityinen avain ja julkinen osoite. Lähetän kolikot Coinbasesta lompakon julkiseen osoitteeseen, ja muutaman sekunnin päästä ne ilmestyvät Bitpay-lompakkooni. Bitcoin-verkko ottaa tästä 13 sentin välityspalkkion, joka valuu verkon louhijoille.

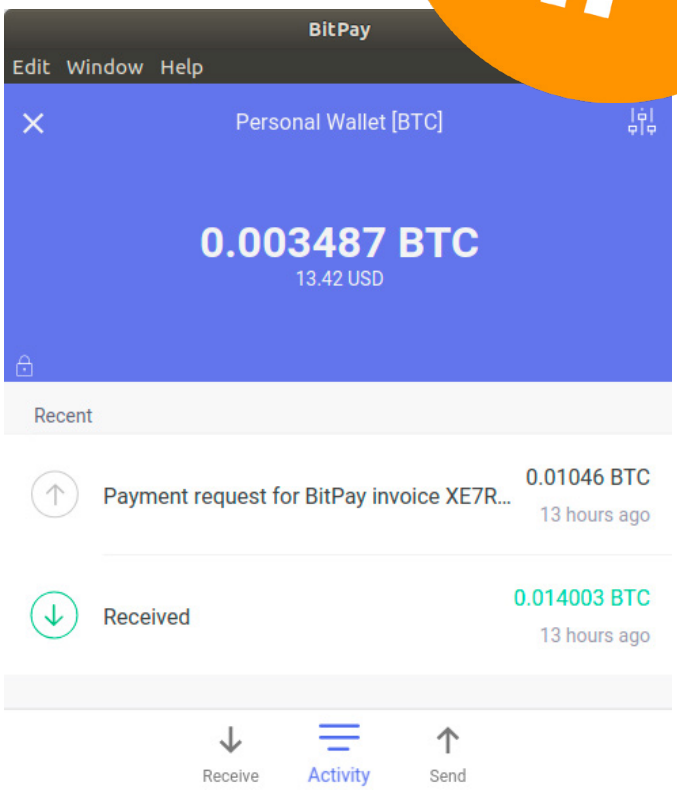
Kirjoitan Namecheapin hallintapaneeliin, että haluan vastaanottaa 25 dollaria Bitpayn kautta, ja saan takaisin pitkän merkkijonon. Kopion tämän Bitpayn asiakasohjelmaan ja lähetän kolikot matkaan. Bitpay ottaa välistä 20 sentin suuruisen palkkion. Parin sekunnin päästä tilausvahvistus ilmestyy Namecheapin paneeliin, ja operaatio on suoritettu onnistuneesti.

Tehtävä suoritettu

Koko hommaan meni alle puoli tuntia. Loppujen lopuksi transaktio meni siis yllättävän sujuvasti. Vaikka uusia transaktioita lisätään Bitcoinin lohkoketjuun vain noin joka kymmenes minuutti, tämä ei näkynyt ulospäin. Minulle näytti siltä kuin siirrot menisivät läpi parissa sekunnissa.

Erilaisiin välityspalkkioihin katosi huomattavan paljon rahaa. Näistä suurin osa liittyi kolikkojen ostoon. Kun kolikot oli kerran ostettu, niiden siirtely eteenpäin oli kohtuullisen halpaa. Pienemmällä välityspalkkiolla suhteessa ostettuihin kolikoihin selviää ostamalla enemmän kolikkoja kerralla.

Perinteinen luottokortti voittaa Bitcoinin vielä sekä käytettävyydessä että (kuluttajalle lankeavissa) siirtokuluissa silloin kun se toimii. Bitcoin tarjoaa kuitenkin varteenotettavan vaihtoehdon, kun perinteiset väylät tökkivät.



Omenaläppärin uusiokäyttö

Kävin alkuvuodesta luokkakokouksessa ja tulin siellä jutelleeksi vanhan koulukaverin kanssa. Puheeksi tuli kannettava tietokone, jonka kävimme lukiossa hänelle ostamassa: Apple iBook G4. Yllättäen hänellä oli kone yhä tallessa! Päätinkin ostaa sen nimelliseen hintaan pois pölyä keräämästä ja tutkia, voisiko sillä vielä tehdä jotain hauskaa.

Teksti: Ronja Koistinen Kuvat: Toni Kortelahti, Ronja Koistinen

Vuosituhannen alun Apple-tietokoneet perustuivat vielä IBM:n PowerPC-suoritinarkkitehtuuriin. Myöhemmin vuonna 2006 Apple vaihtoi suorittimensa Intelin x86:een. PowerPC-pohjaiset iBookit ja PowerBookit korvautuivat tuolloin Intel-pohjaisilla MacBook- ja Macbook Pro -koneilla. Viimeinen PowerPC:tä tukenut Mac OS X -käyttöjärjestelmän versio on 10.5 ”Leopard” – sekään ei tosin enää tue vanhempia G3-suorittimia.

Tämän koneen tekniset ominaisuudet selviävät oheisesta taulukosta.

Malli	iBook G4 1,2 GHz (2004, mallinro A1054)
Suoritin	PowerPC 7447a (G4) 1,2 GHz
Keskusmuisti	256 Mt DDR (max 1,25 Gt)
Kiintolevy	30 Gt Ultra ATA
Optinen asema	Kelkaton DVD-ROM/CD-RW
Näyttö	12,1 ” TFT (1024×768)
Näytönohjain	ATI Mobility Radeon 9200, 32 Mt
Verkkoyhteydet	10/100BASE-T ethernet, Airport Extreme (802.11g), 56k-modeemi
Mitat	3,4 cm × 28,5 cm × 23,0 cm
Paino	2,2 kg
Muuta	2 × USB 2.0, 1 × FireWire 400, BlueTooth 1.1

Ensi töikseni halusin koneeseen enemmän levytilaa, koska 30 gigaa käy nykyään ahtaaksi. Lisäksi vanha 4200 kierrosta minuutissa pyörivä kiekko ei houkuta noin muutenkaan. Tilasin siis netistä mSATA–IDE-adapterin sekä 120 gigan SSD:n.

Koneen avaaminen kiintolevyn vaihtamiseksi oli yllättävänkin monimutkainen projekti, joten en kuvaa sitä tässä tarkemmin, mutta erinomainen iFixit-sivusto tarjoaa kattavan kuvallisen ohjeen[1].

Avaaminen ei mennyt itselläni ihan putkeen. Viime metreillä onnistuin yläkantta nostaessani ottamaan kiinni akkukotelon ohuesta reunasta ja napsauttamaan koneen alumiinirungosta yhden nurkan kokonaan irti.

Korjailua odotellessa päätin, että haluan kokeilla SSD:n toimivuutta ja käyttöjärjestelmän asennusta koneen kuoret auki. (Onneksi ei ole puolisoa, joka hermostuisi siitä, että keittiön pöydällä on viikon verran tietokone levällään odottamassa Unix-leikkejä.)

Massamuistin vaihdon jälkeen kokosin konetta sen verran, että sain kiinni näppäimistön ja touchpadin. Laitoin virrat päälle, ja kone päästi ihastuttavan Applen käynnistysäänen,

Konetta purkaessa akkukotelon kohdalta murtui pala alumiinirunkoa.

Amazonista ostetut mSATA-IDE-adapteri sekä 120 gigan SSD.

mutta hetken päästä jäi hölmistyneenä ihmettelemään puuttuvaa käyttöjärjestelmää.

Uusi käyttöjärjestelmä

Koska Apple ei enää julkaise tällä koneella toimivaa käyttöjärjestelmää, on ajantasainen sellainen saatava jostain muualta. Vaihtoehtoja on muutamia.

Linux tietysti periaatteessa toimii PowerPC-koneilla, mutta jakeluversioissa on nykyään kovin vähän valinnanvaraa. Aikoinaan suosituimpia jakeluita PPC:lle oli Yellow Dog Linux, mutta sen viimeisin julkaisu on vuodelta 2012, eikä sen nettisivukaan enää toimi.

Lubuntu-jakelusta on vielä 16.04-version PPC-asennuskuvatiedosto olemassa[2], ja Ubuntu 16.04:hän on LTS-versio, jolla on tukea jäljellä huhtikuuhun 2021. Samoin Ubuntu MATE -jakelusta on saatavilla 16.04-versio[3]. Googlailemalla ja Facebookin *Linux on PowerPC Macs* -ryhmää[4] selailemalla löytyy myös eritasoisia pilipalijakeluita.

Debianissa 32-bittinen PowerPC on tuettu versioon 8 *”jessie”* asti (tätä kirjoitettaessa uusin vakaa versio on 9 *”stretch”*). Versio 9 tukee edelleen myös 64-bittistä PowerPC:tä (esimerkiksi Applen G5-suorittimet).[5]

Gentoossa vaikuttaa olevan edelleen täysi PowerPC-tuki, myös 32-bittisenä.

Mutta minä suuntasin katseeni siihen käyttöjärjestelmään, jonka vitsaillaan pyörivän vaikka leivänpaahtimessa: NetBSD! Sen kanssa en usko olevan huolta siitä, että seuraavassa versiossa pudotetaan PowerPC-tuki pois. Tukeehan se muun muassa sellaisia hieman jo ikääntyneitä alustoja kuin Amigat, Sparcit ja Alphat – uusimmassakin versiossaan (tätä kirjoitettaessa 8.0).

NetBSD:n asennusohjeet PowerPC-Appleihin ovat myös erittäin kattavat verrattuna useimpiin Linux-jakeluihin.[6]

Asennusmedian boottaaminen

Applen PowerPC-koneiden vastine PC-koneiden BIOSille on Open Firmware. Tämä on emolevyn ROM-piirille tallennettu laiteohjelmisto, jolla kone boottaa. Tässä iBook G4:ssä on Open Firmwaren versio 3, joka osaa lukea käynnistyksen seuraavan vaiheen bootloaderin FAT-, HFS(+)- tai hybridi-HFS-ISO9660-tiedostojärjestelmältä tai ethernetin yli.

Ensimmäiseksi minun pitää siis luoda uudelle puhtaalle SSD:lleni Apple Partition Map[7] -tyyppinen osiotaulu sekä HFS- tai FAT-tyyppinen osio. Osiolle täytyy kopioida bootloader, jonka NetBSD tarjoaa tiedostossa nimeltä `ofwboot.xcf`.

Levyn osioiminen onnistui koneen mukana tulleella OS X:n asennuslevyllä. Käynnistin koneen uudelleen ja syötin sille DVD:n. Käynnistyi asennusohjelma, josta löytyi Disk Utility -niminen ohjelma. Tätä käyttämällä tein SSD:n alkuun 10 gigatavun kokoisen HFS-osion, jonka Open Firmware tarvitsee boottaamiseen. Asensin tälle osiolle myös OS X:n siltä varalta, että tarvitsisin sitä joskus. Kopioin osiolle USB-tikulta OS X:ää käyttäen `ofwboot.xcf`-tiedoston odottamaan tulevaa.

Sitten käynnistin iBookin Open Firmwaren komentokehotteeseen. Tämä tapahtuu käynnistämällä kone kylmiltään ja painamalla käynnistysäänen aikana näppäimiä ⌘–⌥–O–F ja pitämällä ne pohjaan painettuna, kunnes komentokehote ilmestyy.

Yritin alkuun bootata NetBSD:n asennusta USB-tikulta. Vaikka tämän internet-lähteiden mukaan pitäisi olla mahdollista, en saanut tätä onnistumaan millään ilveellä. Siispä poltin asennuskuvan CD-RW:lle.

Käynnistin asennuksen Open Firmwaren komentokehotteesta komennolla:

```
boot cd:,\ofwboot.xcf netbsd.macppc
```

Levyn osioiminen

Asennus boottaa ja kysyy ensimmäiseksi terminaalin tyyppiä. Oletusvastaus on ”vt100”, joka on ihan toimiva, joten painan vain enteriä. Kysymykseen ”(I)nstall, (S)hell or (H)alt ?” vastaan S, jotta pääsen asennuksen komentokehotteeseen.

Seuraavaksi pitää luoda loput osiot, joille NetBSD asennetaan. Asennusmanuaali suosittelee luomaan ainakin root- ja swap-osiot sekä HFS-osion, jonka kautta tiedostoja voi siirtää NetBSD:n ja OS X:n välillä.

Levyn osiointi on BSD-järjestelmissä varsin erilaista kuin esimerkiksi Windowsissa tai Linuxissa. NetBSD:n macppc-versiolle on aiheesta laaja ohje[8]. Karkeasti ottaen BSD-käyttöjärjestelmissä levyn raakaosioiden sisään laaditaan vielä toinen *disklabel*-nimeä kantava osiointiskeema, jossa osiot luetellaan kirjaimilla a–h.

Ensimmäiseksi selvitän `dmesg`-tulosteesta, että SSD-levyni nimi on `/dev/wd0`. Avaan pdisk-osiointityökalun komennolla `pdisk /dev/wd0c`. Lopun ”c” on disklabel, joka viittaa koko wd0-lohkolaitteeseen (*block device*).

Komennan pdiskin kehotteeseen C (iso kirjain) eli ”create new partition with type also specified”. Teen osion, joka alkaa tyhjästä osiosta 4 (4p), on kaksi gigatavua pitkä (2g), saa nimen

Vasemmassa alakulmassa kiintolevyn tilalle vaihdettu SSD adapterikotelossaan. Yläreunassa näkyy myös koneen valmistuspäivän leima: 24. syyskuuta 2004.

”swap” ja on tyyppiä ”Apple_UNIX_SVR2” (asennusmanuaali puhuu tästä tyyppinä ”A/UX”). Sitten pdisk kysyy, minkä slice-kirjaimen haluan antaa osiolle. Vastaan b, joka on tarkoitettu swap-osioille.

Seuraavaksi teen root-osion seuraavin määrittein: alkaa tyhjästä osiosta 5 (5p), on koko tyhjän osion pituinen (5p), nimeksi ”root”, tyypiksi sama kuin yllä, slice-kirjaimeksi a (root partition).

Lopuksi kirjoitetaan uusi osiotaulu levylle komennolla w ja poistutaan pdisk-ohjelmasta komennolla q.

NetBSD:n asentaminen

Nyt kun SSD on osioitu, siirryn asennusohjelmaan komennolla `sysinst`.

Valitsen ensimmäisestä valikosta kieleksi englannin ja seuraavasta vaihtoehdon ”a: Install NetBSD to hard disk”. Varmistuskysymyksen jälkeen asennus kysyy, mille levylle NetBSD asennetaan. Valitsen kohdan a: wd0. Seuraavasta valikosta valitsen b: Use existing partition sizes. (Vaihtoehdosta a olisin voinut esimerkiksi korvamerkitä /home-disklabelille tietyn verran tilaa, mutta minulle kelpaa, että kaikki tila on suoraan root-osion hallussa.)

Asennus näyttää nykyisen osiotilanteen. Valitsen x: Partition sizes ok. Seuraavaksi kysytään nimeä valitsemalleni levylle. Tällä ei liene suurta merkitystä – kirjoitan ”SSD”.

Sitten vielä varmistus: Shall we continue? Yes.

Asennus alustaa root-osiolle FFS-tiedostojärjestelmän. Sitten kysytään, millaisen asennuksen haluan. Valitsen a: Full installation. Asennan lähteestä a: CD-ROM / DVD, sillä tällä hetkellä mikään verkkolaite ei ole kytketty saati konfiguroitu.

Sitten voisi vielä valita CD-aseman ja pakettien polut asemassa olevalla levyllä. Oletukset näyttävät hyvältä, joten valitsen x: Continue.

Tiedostojen kopioinnin jälkeen tarjotaan vielä valikko, jossa voi tehdä erinäisiä asetuksia ennen tietokoneen uudelleenkännistystä. Käyn asettamassa aikavyöhykkeen ja rootin salasanan, sitten valitsen x: Finished configuring.

Lyhyen uuden järjestelmän asetuksia koskevan ohjepläjäyksen jälkeen pääsen takaisin alkuvalikkoon. Vaitsen d: Reboot the computer.

Asennetun käyttöjärjestelmän käynnistäminen

NetBSD:n käynnistäminen kiintolevyltä ei alkuun onnistunut Open Firmware -kehotteesta asennusohjeen mukaisesti[9]. Bootloader löytää antamallani komennolla kernelin, mutta jumiutuu sitä ladatessaan. Pyysin apua postituslistalta ja sain vastauksen, jonka mukaan bootloader ei jostain syystä aina osaa käynnistää kerneliä, jos se on liian suuri. Sain ohjeeksi pakata kernelin gzipillä.

Ratkaisin ongelman kopioimalla OS X:n puolella asennus-CD:ltä tiedoston `/binaries/kernels/netbsd-GENERIC.gz` OS X -osion juureen. Tämän jälkeen bootti onnistui komennolla:

```
boot hd:,\ofwboot.xcf netbsd-GENERIC.gz
```

Asetuksia ja asennuksia

Ensimmäiseksi luon käyttäjätunnuksen itselleni:

```
# useradd -g users -m ronja
```

Näppäimistöasettelu täytyy vaihtaa suomalaiseksi. Se onnistuu lisäämällä rivi ”encoding fi” tiedostoon `/etc/wscons.conf` ja ”wscons=YES” tiedostoon `/etc/rc.conf`. Muutos tulee näillä toimenpiteillä voimaan seuraavassa bootissa. Saman tien sen saa käyttöön komennolla:

```
# wsconsctl -k -w encoding=fi
```

Säädetään vähän myös verkkoasetuksia. Lisätään tiedostoon `/etc/rc.conf` rivit `hostname="koneennimi"` sekä `dhcpcd=YES`. rc.conf ajetaan vain bootin yhteydessä, joten jotta verkkoyhteys saadaan päälle heti, ajetaan `dhcpcd`-komento saman tien root-konsolissa.

NetBSD:ssä on kolme erilaista tapaa asentaa ohjelmia: `pkgsrc`[10] lähdekoodista asentamiseen (vrt. Open- ja FreeBSD:ssä ”ports”), `pkg_add` binääripaketteja varten sekä `pkgin`. Jälkimmäinen on näistä edistynein, mutta sitä ei ole saatavilla valmiiksi asennettuna. (Tai ehkä se olisi, jos olisin osannut valita asennusvaiheessa oikeat valinnat.) Joudumme ensin asentamaan sen pkgsrc-reittiä.

Pkgsrc-tiedostot on ensin haettava ja purettava oikeaan polkuun:

```
# ftp ftp://ftp.NetBSD.org/pub/pkgsrc/pkgsrc-2019Q1/pkgsrc.tar.gz
# tar -xzf pkgsrc.tar.gz -C /usr
```

Sitten alustamme pkgsrc-ympäristön:

```
# cd /usr/pkgsrc/bootstrap
# ./bootstrap
```

Pkgin asentuu seuraavin komennoin:

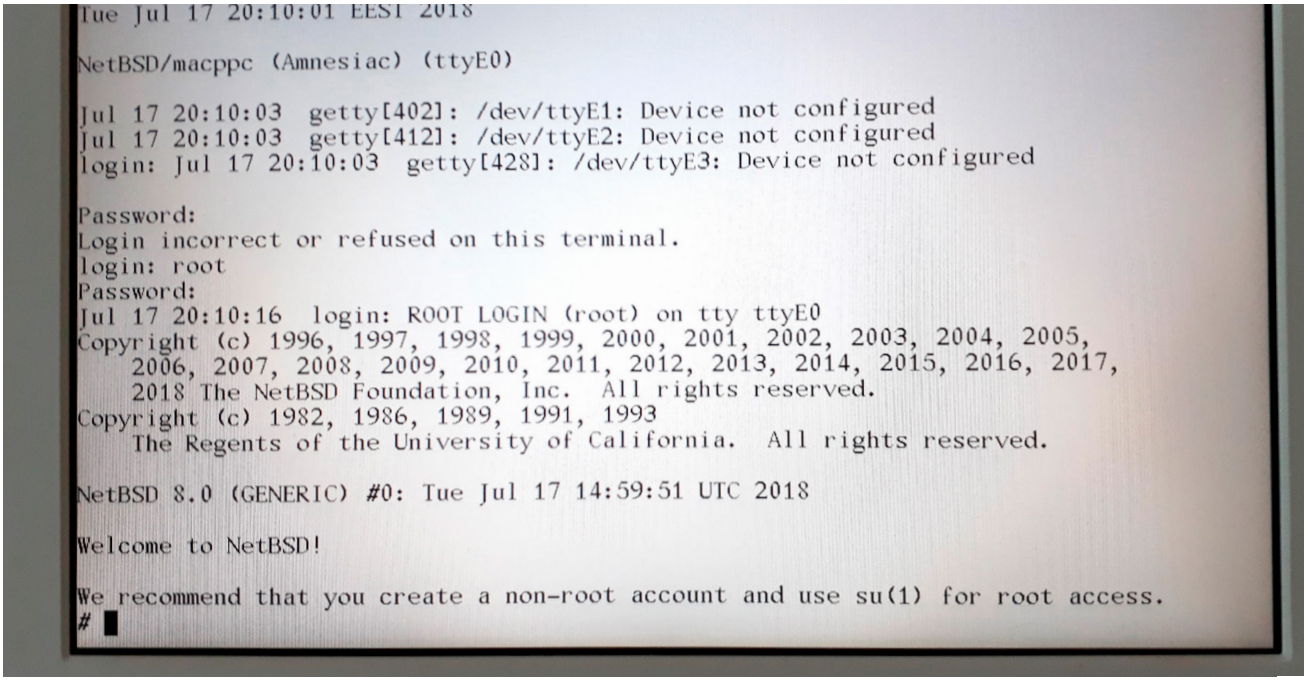
```
# cd /usr/pkgsrc/pkgtools/pkgin
# make
# make install
```

Graafinen käyttöliittymä

Nyt kun meillä on pkgin, voimme asentaa sillä ohjelmistoja. Ensiksi asensin sellaisia mukavuuksia kuin Vim, htop, OpenNTPD, Bash ja rsync. Mutta jotta koneen kanssa pääsee kunnolla hommiin, tarvitaan X-Window ja jokin ikkunointiympäristö.

Asennetaan Openbox-ikkunointijärjestelmä ja sen konfigurointityökalu:

```
# pkgin install openbox obconf
```



Onnistuneen bootin jälkeen ensimmäinen kirjautuminen root-käyttäjänä.

Lisätään tavallisen käyttäjämme kotihakemistossa olevaan .xinitrc-tiedostoon rivi ”openbox”. Sen jälkeen saamme graafisen X-Window-ympäristön käyntiin ajamalla käyttäjänämme komennon startx.

Ruutuun ilmestyy musta tausta ja hiiren kursori. Valitettavasti tässä vaiheessa pitää ottaa käyttöön ulkoinen USB-hiiri, sillä X-Window on erittäin penseä käytettävä iBookin yksipainikkeisella touchpadilla. Openboxin sovellusvalikko nimittäin aukeaa klikkaamalla työpöytää hiiren oikealla napilla.

Toimivan web-selaimen saaminen muulle kuin x86-, amd64- tai ARM-arkkitehtuurille alkaa nykypäivänä olla hankalaa ainakin GNU/Linux- ja BSD-käyttöjärjestelmille. Firefoxista tai Chromiumista on turha haaveilla, joten asensin iBookiin testailuun Dillo- ja Netsurf-nimiset selaimet. Kumpikaan niistä ei oikein osaa piirtää ainakaan Skrollin kotisivuja kunnolla, ja molemmilla tuntuu olevan myös suuria vaikeuksia verifioida SSL-sertifikaatteja. Terminaaliselaimet ELinks ja w3m tuntuvat olevan toimivimmat.

Asennusinto alkaa tältä erää hiipua, joten kone pääsköön joksikin aikaa lepäämään. Puuhaa kuitenkin vielä riittää, ja jos tämä artikkeli saa kiinnostunutta palautetta, näihin voidaan vielä Skrollin sivuilla palata:

1. Langaton verkko olisi hauska virittää käyttöön – tukea Airport Extreme -kortille pitäisi olla.
2. Ääniä ei ole vielä edes kokeiltu.
3. Tällä hetkellä NetBSD:n puolelle boottaaminen vaatii Open Firmwaren puolelta naputtelemisen. Olisi kiva jaksaa vielä muuttaa Open Firmwaren oletusboottausta siten, että NetBSD käynnistyy ilman kommervenkkejä.
4. iBookin emolevyllä ei ole reaaliaikakellon varmistusparistoa samalla tavalla kuin PC-koneissa, vaan kellonaikaa ylläpidetään pelkästään koneen akulla. Akku ei kuitenkaan ilmeisesti enää ole entisensä, joten NetBSD tulostaa joka bootissa huolestuneita viestejä kellonajan tilasta. Tälle olisi ehkä hyvä yrittää tehdä jotain.
5. Youtubea saattaisi voida katsoa youtube-dl:n sekä vlc-soittimen yhdistelmällä, kunhan vain imuroi videoista maltillisen kokoiset versiot, joita G4 jaksaa jotenkuten purkaa.
6. X:ssä pitäisi vielä ottaa käyttöön suomalainen näppäinasettelu ja UTF-8. Nämä kyllä sinänsä hoituvat laittamalla `.xinitrc`-tiedostoon rivit `export LC_ALL=en_US.UTF-8` ja `setxkbmap fi`. Ongelmia tuleekin siitä, että <-näppäin, josta Mac-näppäimistöllä Altin kera pitäisi saada putki eli |, alkaakin tulostaa §- ja ö-merkkiä UTF-8-localessa. Tarvitaan siis jotain erikoisloitsuja vielä setxkb-mapille tai kustomoitu xmodmap, jotta Mac-näppäimistön erikoisuudet saadaan toimimaan. Tähän ei nyt riittänyt hermoja lehden deadlinen puitteissa.
7. Jos BSD-nyhjääminen alkaa hatuttaa liikaa, voidaan vaihtaa koko hoito MorphOS:ään.

Viitteet

[1] *ifixit.com/Guide/iBook+G4+12-Inch+800+MHz-1.2+GHz+Hard+Drive+Replacement/166*
[2] *lubuntu.me/downloads/*
[3] *ubuntu-mate.org/download/*
[4] *www.facebook.com/groups/ppclinux/*
[5] *www.debian.org/ports/powerpc/*
[6] *cdn.netbsd.org/pub/NetBSD/NetBSD-8.0/macppc/INSTALL.html*
[7] *en.wikipedia.org/wiki/Apple_Partition_Map*
[8] *www.netbsd.org/ports/macppc/partitioning.html*
[9] *gnats.netbsd.org/cgi-bin/query-pr-single.pl?number=54096*
[10] *www.netbsd.org/docs/pkgsrc/index.html*

Digiarkeologian alkeet

Näin sinustakin tulee KASETTILAMERI

Kasettilameri oli aikanaan haukkumanimi vähän noloille kasettipelien pelaajille ja vaihtelijoille – nykyisin se on kunniakas arvonimi vanhan datan pelastajille. Kasettilamerit-ryhmän perustaja kertoo nyt, miten sinustakin tulee C-kasettitieteen digiarkeologi.

Teksti: Jarkko Lehti
Kuvat: Emma Kantanen, Jarkko Lehti, Mitol Meerna, Janne Sirén

Magneettinen media nimenä saattaa nykypäivänä merkitä lähinnä hölynpölyä tai menneiden aikojen tallennusmedioita. Se viimeinenkin magneettisen median linnake, kiintolevy, on korvautunut pitkälti SSD-muistipiireillä.

1980-luvulla magneettinen media oli kuitenkin hiljalleen syrjäyttänyt mekaanisen median (LP-levy, ks. sivu 60) ja optinen media (CD-levy) oli vasta nousemassa. Magneettinen media oli siis tuolloin valtavirtaa ja myös tietokoneenkäyttäjän arkipäivää. Kotimikroilijatkin tallensivat tuotoksensa ja kopiopelinsä niin kutsutuille C-kaseteille (muitakin kasettikokoja oli) ja levykkeille.

Tutustumme nyt kasettien nauhamuotoiseen tiedon tallennukseen tai oikeastaan sen lukemiseen vanhan datan pelastamisen näkökulmasta. Samat periaatteet sopivat pitkälti myös levykkeille. Nekin ovat nimittäin pohjimmiltaan vain muovikalvoja, joihin on liimattu kiinni magneettinen pinnoite, johon hyötysignaalia voidaan taltioida.

Kasettitieteen perusopinnot

Aluksi on syytä tutustua magneettisten nauhojen valmistustapaan sekä nauhalle tallentamisen ja sieltä lukemisen teoriaan.

Nauha on muovikalvoa, jonka päälle on liimattu itse magneettinen materiaali, johon tietoa voidaan tallentaa. Magneettinen materiaali koostuu mikroskooppisista magneettisista hiukkasista, jotka halvimmilla tavallisilla nauhoilla ovat rautaoksidia eli ruostetta. Usein nauhat ovatkin ruosteen värisiä, mutta toisinaan näkee peruskasetteja myös mustina, tällöin massaan on lisätty yleensä hiiltä väriaineeksi. Jokaisella valmistajalla kuitenkin lienee ollut oma reseptinsä nauhan valmistukseen.

Muovikalvon päälle on siis kiinnitetty mas-

saa, jossa on metalliainesosa ja liima. Tämä kerros on hyvin ohut. Tyypillisen C-kasetin nauhapaksuus onkin 16 mikrometrin luokkaa, ja kun standardi nauhanopeus on 4,76 cm/s, tavallisen 60 minuutin nauhan pituus kelalla on huikeat noin 85 metriä. Jos joku tarkasti laskelmani ja riemuissaan huomasi virheellistä tietoa, kannattaa muistaa, että 60 minuutin C-kasetin pituus on 30 min/puoli. Jotkut kasetinvalmistajat myös ilmoittivat nauhan pituuden metreissä: tavallisimmin olen nähnyt 60 minuutin kasetilla merkinnän 90 m.

C-kasetit ovat yleensä pituudeltaan 60 ja 90 minuuttia, joskin aikanaan oli saatavilla myös 120 minuutin nauhoja, mutta niiden nauhan paksuus oli jo niin ohutta, että nauha venyi herkästi käytössä. Nauhan piti siis olla ohuempaa, että sitä mahtui enemmän c-kasetin kuorien sisään. Tämä ominaisuus ei mitenkään lisännyt 120 minuutin nauhojen luotettavuutta. Lisäksi myytiin eri pituisia lyhyempiä ”datanauhoja”, joiden pituus vaihteli paljonkin. Minulla ei ole tiedossa olivatko nämä datanauhat paksumpaa materiaalia tai muuten tekniseltä laadultaan parempia. Käytännössä niihin tarttui data samalla tavalla kuin äänikasetteihinkin.

Kasetti itsessään koostuu kotelosta, kahdesta kelasta ja aukosta, jonka kautta nauhaa kulkee kelalta toiselle. Nauhan takana aukon keskellä on jousella viritetty huopatyyny, jonka tehtävä on painaa nauhaa äänipäätä vasten. Kasetin kotelon sisällä on tavallisesti myös muoviset kalvot, joiden tarkoitus on vähentää kitkaa nauhakelojen pyöriessä. Kaseteissa on usein myös ikkuna, josta voi nopeasti nähdä, missä kohtaa nauha on ajallisesti menossa.

Kotelon tehtävä on pitää lika ulkopuolella, mutta riippuen nauhojen säilytyksestä kotelo ei toimi aukottomasti tehtävässään. Nauhan sisälle pääsee esimerkiksi kosteus liiankin helposti.

Nauhaa toistavan nauhurin tehtävä on pyörittää kasettia tasaisella 4,7 cm/s nopeudella, ja äänipään tulisi olla suorassa linjassa nauhan kanssa. Varsinkin äänipään oikean kulman (atsimuutti, *azimuth*) kanssa on usein ongelmia, sillä maksimaalisen signaalin voimakkuuden ja oikeanlaisen aaltomuodon saamiseksi sen pitää vastata säätöä, joka kasetin kirjoittaneessa asemassa oli. Varsinkin Commodore 64:n nauhojen kanssa oli varsin tavallista, että säätöruuvi tuli tutuksi jokaisen kasetin ollessa hieman eri säädöllä. Pahimmillaan ruuvia piti veivata myös kesken kasetin puolen, kun ohjelmia oli kopioitu kasetilta, jossa säätö oli ollut toinen.

Se miksi säätö oli lähes jokaisella käyttäjällä erilainen, johtui yksinkertaisesti C64:n kasettiasemien todella huonosta laadusta: tehtaalla ei liene juuri aikaa kulutettu laitteiden kalibrointiin, vaan niitä tuutattiin maailmalle sen minkä ehdittiin. On vaikea löytää 80-luvun elänyt Commodore 64 -käyttäjä, joka ei muistaisi kauhulla kasetilta latausta ja tarinointia loputtomasta säätämisestä. Tässäkin asiassa totuus on usein tarua ihmeellisempää. Loppupeleissä harvoilla taisi olla edes käsitystä, miksi sitä ruuvia edes ruuvattiin. Koitan avata hieman näitä ongelmia myöhemmin.

Kun nauha pyörii nauhurissa tasaisella nopeudella ja äänipään kulma on optimaalinen, nauhalle tallennetut magneettikentän vaihtelut saavat aikaan sähkövirran äänipään sisällä olevalla kelalla. Nämä heikot jännitteen vaihtelut vahvistetaan äärimmilleen monessa eri vaiheessa, jonka lopputulos on +5 ja -5 voltin välillä vaihteleva kanttiaalto, joka kuvastaa bittejä 1 ja 0. Jos äärimmäinen vahvistus jätetään väliin, saadaan signaali, joka muistuttaa enemmän tai vähemmän siniaaltoja – esimerkiksi kun datakasetti luetaan tavallisella kasettisoittimella, joka on tarkoitettu musiikin toistamiseen. Jos taasen nauhalle halutaan tallentaa tietoa, se tehdäänkin päinvastaisesti. Ensin nauha esimagnetoidaan tyhjennyspäällä, jonka jälkeen äänipäähän ajetaan voimakas signaali, joka kirjoittaa magneettikentän vaihtelut ohikulkevalle nauhalle.

Oikeasti nauhalle tallennus on hieman monimutkaisempaa, varsinkin jos tarkoitus on tallentaa hifiääntä, mutta yksinkertaistettuna prosessi menee näin.

Äänipäällä on sähköinen kela, kuten jo aiemmin kerrottiin. Pelkkä kela ei vielä riitä, vaan äänipään tarvitsee myös niin sanotun ilmaraon signaalien tulkkaamiseen. Mitä kapeampi ilmarako on, sitä korkeampia taajuuksia saadaan kaapattua. Hifilaitteiden alkuaikoina äänipäiden laadussa olikin eroja eri valmistajien prosessista riippuen, mutta viimeistään kun laserleikkaus tuli teollisuuteen, äänipäistä tuli hyvinkin tasalaatuisia, mitä ilmaraon kokoon tulee. Tämän lisäksi äänipään valmistusmateriaali vaikuttaa sen laadukkuuteen lähinnä kulumisen muodossa. Halvimmat päät olivat heikompaa metallia ja kuluivat kuopalle kovassa käytössä. Helpoin tapa tarkistaa äänipään kunto onkin kokeilla kynnen päällä, tuntuuko siinä porrasta nauhan kulkureitin reunoilla. Jos porras tuntuu, äänipää on kulunut liikaa ja on vaihtokunnossa.

Nauha kulkee siis äänipään ilmaraon päältä, ja jos ilmarako ei ole tarkasti linjassa nauhan kanssa, tulee ongelmia, sillä magneettinen materiaali nauhan päällä ei olekaan magnetoitunut suorassa linjassa. Jotta alkuperäinen signaali saadaan luettua täydellä voimakkuudella ilman vaiheongelmia, äänipää pitää saada samaan kulmaan kuin kasettia tallennettaessa. Kun luemme ääntä tavallisella äänentoistoon tarkoitetulla soittimella, meillä ei ole oikein mitään järkevää mahdollisuutta päätellä signaalin laatua prosessin aikana.

Lisäksi jos haluaisimme lukea datan audioksi pieteetillä, äänipäätä pitäisi voida säätää koko ajan, mikä tekee soittimesta musiikkikäyttöön kelvotto-

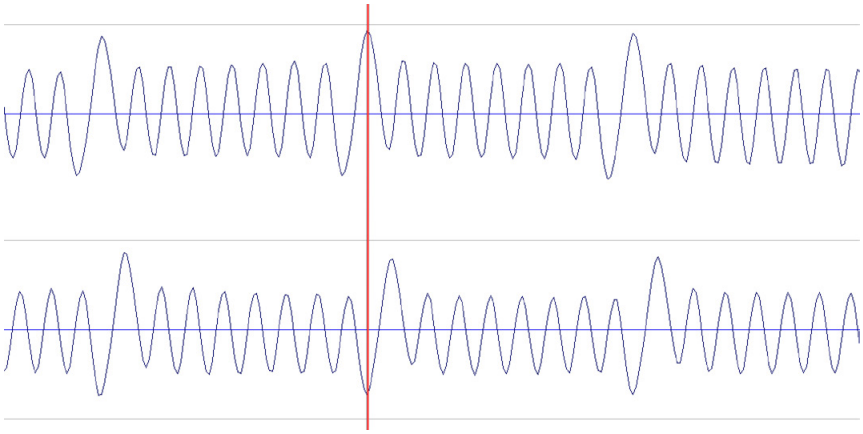
man. Jos äänipään asentoa mennään muuttamaan, se pitäisi kalibroida jokaisen säädön jälkeen oikeaan kulmaan musiikkikäyttöä varten. Tästä syystä olisi parempi olla aivan eri laite musiikille kuin mahdolliseen datakasettien lukemiseen, jos äänipään säätöön halutaan mennä kajoamaan.

C-kasettinauhassa on tavallisesti neljä raitaa, mutta on myös olemassa kahdeksanraitaisia C-kasettinauhureita, jotka ovat mielenkiintoisia teknisiä kuriositeetteja lähinnä kotistudiokäyttöön. Standardilla nauhalla on kaksi raitaa kasetin puolta kohden: vasen ja oikea kanava. Monolaitteet tallentavat saman signaalin molemmille kanaville, jolloin äänipää on kahden raidan levyinen. Tämä leveämpi äänipää on yksi syy, miksi atsimuuttisäädön pitää olla niin tarkasti oikein. Mitä kapeampi äänipään raita on, sitä paremmin se sietää väärää atsimuuttia. Luettaessa kasettia hifidekillä väärällä säädöllä ei siis ole niin isoa merkitystä, sillä voimme valita paremmin paremman signaalin kaapanneen kanavan jatkokäyttöön.

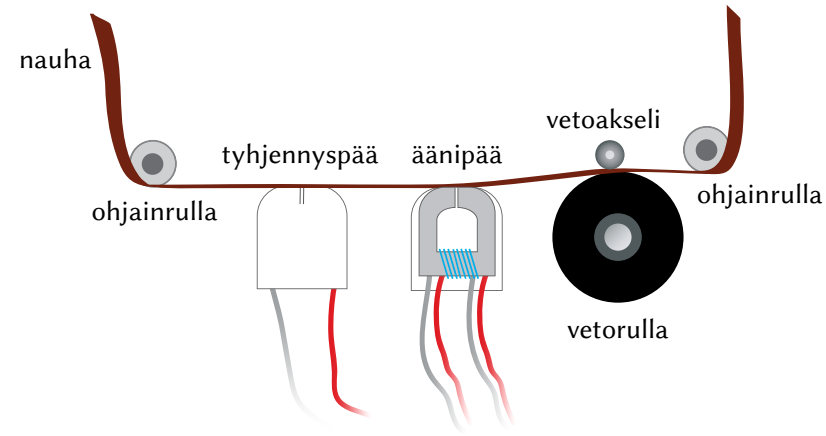
Kasetin lukeminen hifidekillä

Oheisesta kuvasta näemme, miten tallentavassa asemassa on atsimuutti ollut pielessä verrattuna lukevaan kasettidekkiin. Jos olisimme lukeneet signaalin leveämmällä mono-äänipäällä, kanavat olisivat sekoittuneet yhteen, eikä signaali enää olisi muistuttanut alkuperäistä, jolloin lataus olisi muuttunut mahdottomaksi. Kun valitsemme vain joko vasemman tai oikean kanavan, saamme alkuperäisen signaalin muodon paremmin talteen.

Kasetilta dataa lukiessa stereodekillä ei siis saa mennä miksaamaan kanavia yhteen, vaan on valittava paremman signaalin tallentanut kanava! Kun olemme saaneet valittua paremman kanavan kahdesta, hylkäämme toisen ja muunnamme tiedoston monoksi, jossa on vain yksi ääniraita. Signaalin analogisen luonteen suhteen siinä näkyy usein kaiken maailman heittoja, piikkejä ja häipymiä. Näille voi koittaa tehdä jotain käsin äänieditorissa tai kokeilla onnea tallentamalla wav-tiedoston ja ajamalla sen muunnosohjelman läpi. Jos kävi todella hyvä tuuri, signaali oli riittävän ehjää ja pääsemme pelaaman pelastettavaa peliä.



Kuvassa näkyy vasen ja oikea äänikanava luettuna kasetilta, ylempänä vasen ja alempana oikea. Punainen viiva näyttää, miten ylempi ja alempi kanava ovat puoli aaltoa eri vaiheessa.



Nauharata: Kasetin ohjainrullat (guide rollers) ja nauha sekä kasettiaseman tyhjennyspää (erase head), äänipää (play/record head), vetoakseli (capstan) ja vetorulla (pinch roller).

Tosimaailmassa signaali vaatii hieman jatkokäsittelyä. Turbo tape -kaseteissa hyötysignaali on noin kolmen kilohertsin paikkeilla, eikä se oikeastaan muillakaan latureilla mainittavasti eroa siitä. Signaalille kannattaa siis tehdä heti ylipäästösuodatus, jossa kaikki alle 500 hertsin äänet suodatetaan pois. Tämä poistaa muun muassa 50 hertsin hurinat, suurimmat napsujen aiheuttamat signaalin heittelyt ynnä muuta. Jos haluaa hienostella, voi samalla suodattaa korkeammat turhat taajuuskomponentit pois,

mutta tästä ei yleensä ole mainittavaa hyötyä. Sopiva arvo alipäästösuodattimelle on esimerkiksi kuusi kilohertsiä – tämän yli menevät taajuudet kannattaa suodattaa pois.

Tiivistettynä: kun kasettisignaalin ajaa 500 Hz:n–6 KHz:n kaistanpäästösuodattimen läpi, saadaan signaalista enemmän toivotun muotoinen ohjelmistojen pureskelua varten. Vielä on yksi todella tärkeä seikka jäljellä, nimittäin signaalin vaihe. Esimerkiksi C64 lukee signaalista vain nousevaa reunaa. Jos signaali on invertoitunut, bitit menevätkin nurinpäin ja ajoitukset sekoavat, eikä mikään luonnollisesti toimi. Onkin hyvin tärkeää kokeilla tulkata signaali myös invertoituna. Audiokäytössä ei käytännössä ole mitään merkitystä, kumminpäin signaalin vaihe tulee ulos soittimesta, joten siihen on vain harvoin äänentoistolaitteistoissa kiinnitetty huomiota. Datakäytössä taas on elintärkeää, että signaali on oikeinpäin, jotta se on luettavissa. Tämän lisäksi päänvaivaa aiheuttavat äänikorttien sisääntulot, ne kun tuntuvat olevat miten päin sattuu signaalin vaiheen suhteen. Tunne siis oman laitteistosi oikut ja tee muistiinpanoja!

Miksi se oli niin vaikeaa?

En alkuun ymmärtänyt ollenkaan ihmisten valitusta kasetilta lataamisesta, miten se oli ihan onnenkauppaa, ja kuinka säätömeisselistä nähtiin jopa painajaisia. Itse keräsin kokoelman huolellisesti kopioimalla kaiken käsin ja usein korjasin säädönkin omalle kasetille tallentaessa, jolloin säädöt pysyivät yhdenmukaisesti samana saman kasetin ohjelmilla.

Nyt vuosien kasettilamerointikokemuksella minulle onkin selvinnyt, että suurin ongelma aikanaan on ollut pelien kopiointi niin sanotuilla kaksipesäisillä kasettisoittimilla. Signaalin taso on usein todella huono, signaali huojuu ja nauhan kulku kopioidessa on ollut epätarkkaa (ilmeisesti käytetty kaksinkertaista nopeutusta kopioinnissa). Tämän lisäksi eteen on tullut alkuperäisten kopioiden lisäksi turbokasettipelejä kopioituna kaksipesäsoittimella – niiden toimivuus on usein ihan onnenkauppaa. Ei ihme, että aikanaan muodostui legandaksi kasetilta lataamisen vaikeus, vaikka syy oli vain ihmisten laiskuus ja tietämättömyys siitä, mikseivät ne toimi.

Vinkkejä hankaliin kasetteihin

Joskus lameroitava kasetti on heikossa kunnossa: kotelon ikkuna rikki ja niin edelleen. Tällöin kannattaa harkita nauhan siirtämistä uuteen parempaan koteloon. Kaseteissa on toisinaan ruuvit, jolloin siirto onnistuu helposti, toisinaan taas ne on liimattu kiinni. Liimatut kasetit aukeavat lievällä väkivallalla kulmista kääntelemällä, jolloin liimaukset antavat räksähdellen periksi. Nurkissa kannattaa siis pitää hyväkuntoisia koteloita, joissa on ruuvit näitä tapauksia varten, jotta käsiteltävä kasetti saadaan puhtaisiin kuoriin. Eteen tulee säännöllisesti myös nauhoja, jotka eivät tahdo pyöriä kunnolla. Tavaksi onkin tullut jo kelata nauha päästä päähän ennen lameroinnin aloittamista, jolloin vuosia kelalla ollut nauha hieman tuulettuu ja kelautuu uudelleen ja mahdollisesti alkaa pyöriä paremmin.

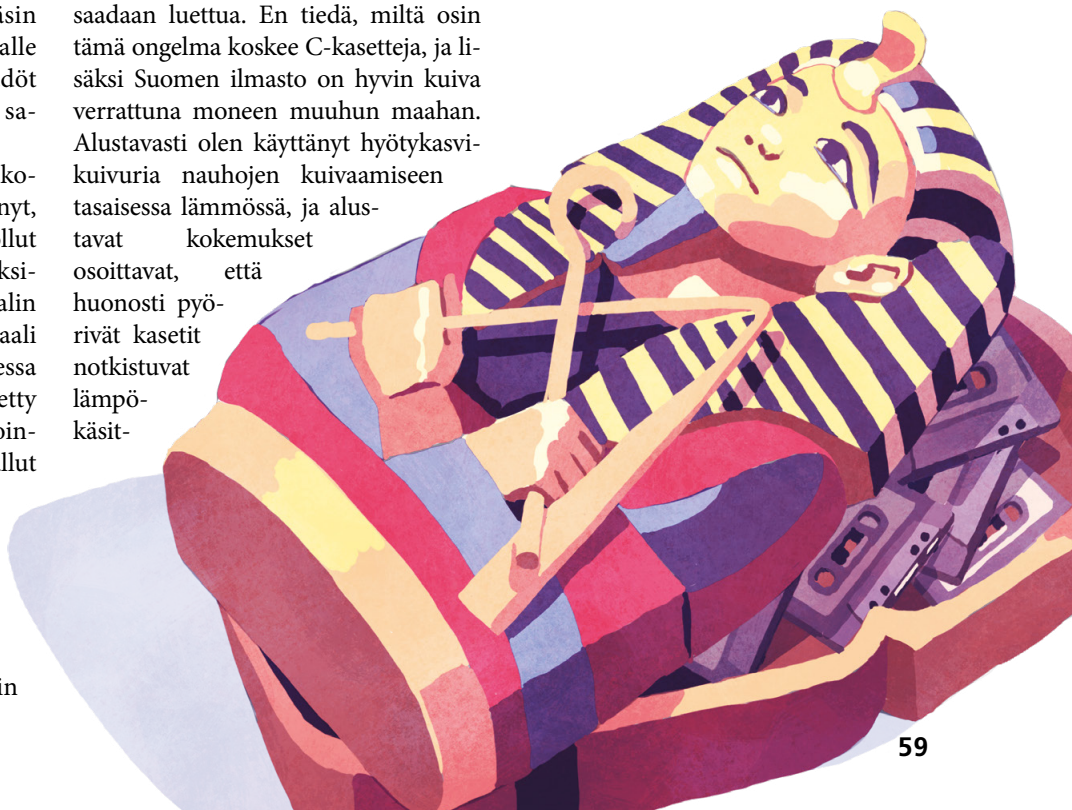
Viime aikoina olemme tutkineet myös ”tahmean nauhan syndroomaa”, jossa lähinnä isompien kelanauhureiden nauhat alkavat mekaanisesti hajoamaan liiman pettäessä, jolloin magneettihiukkaset irtoavat muovikalvolta. Näihin internetin mukaan toimii lääkkeeksi useamman tunnin lämpöhoito noin 50–60 asteen lämmössä, jolloin liimasta pitäisi kertyneen kosteuden poistua ja tarttuvuus palautua, jotta nauha saadaan luettua. En tiedä, miltä osin tämä ongelma koskee C-kasetteja, ja lisäksi Suomen ilmasto on hyvin kuiva verrattuna moneen muuhun maahan. Alustavasti olen käyttänyt hyötykasvikuivuria nauhojen kuivaamiseen tasaisessa lämmössä, ja alustavat kokemukset osoittavat, että huonosti pyörivät kasetit notkistuvat lämpökäsittelyllä. Onko kyse sitten lämpölaajenemisesta, kosteuden poistumisesta tai nauhan tahmeuden poistumisesta, on vielä jäänyt epäselväksi, mutta tätä asiaa on tarkoitus jatkossa tutkia tarkemmin lisää.

Lopuksi tärkein eli puhtaus! Puhdista ennen jokaista kasettia kaikki nauharadan komponentit isopropanolilla. Tyhjennyspää, äänipää, vetoakseli ja rulla. Vetorullasta irtoaa aina mustaa ”likaa”, joten sitä ei kannata hinkata liiaksi, ettei kumi kuivu. Jos esimerkiksi vetoakselilla on jotain tahmeaa likaa, se voi imaista nauhan itseensä kiinni ja lopputuloksena nauha kelautuu akselin ympärille. Lisäksi likainen äänipää ei pysty kaappaamaan signaalia kunnolla, joten puhtaudesta kannattaa pitää huolta. Hyvä tapa on myös puhdistaa nauhuri jokaisen kasetin jälkeen, jolloin lika ei pääse ajan myötä pinttymään.

Kaiken tämän voi myös ulkoistaa Kasettilamereiden asiantuntijoille. Eli jos sinulla on nurkissa datakasetteja, joita et saa itse luettua talteen, ota rohkeasti yhteyttä Kasettilamereihin, ja katsotaan mitä voimme asialle tehdä: www.kasettilamerit.fi.

Teimme aiheesta myös videon Skrollin YouTube-kanavalle: youtube.com/c/skrollimagazine

Lisää Kasettilamereiden toiminnasta ja vanhan datan pelastamisesta muun muassa Skrolleissa 2013.1 ja 2017.4, pdf-lehdet: skrolli.fi/numerot.



LP-ROM?

Dataraita vinyylilevyllä

Teksti ja kuvat: Janne Sirén

Jos ohjelmaa ei saa sovelluskaupasta tai pakettivarastosta, se jää nykyisin monelta käyttämättä. Kotitietotekniikan alkuhämärissä oltiin totuttu työläämpiin jakelutapoihin. Ohjelmia levitettiin BBSien tiedostoalueilla, myytiin kivijalkakaupoissa ja lähetettiin Suomen radiossa. Epätoivoisimmat yrittivät tilata niitä postimyynnillä Komentokeskuksesta.

Cd-rom-tekniikan myötä musiikkicd:illä nähtiin hybridi-cd-rom-dataa ja varoitus, ettei dataraitaa saa soittaa musiikkisoittimessa – kaiuttimet hajoavat. Ohjelmia tosin sisällytettiin musiikkilevyille jo ennen cd-rom-aikaa.

Levysoittimesta -asema

Yksi esimerkki on viime vuonna kuolleen punk-muusikko **Pete Shelleyn** *XL1* (1983). *XL1*:n alkuperäisellä LP:llä on ZX Spectrum -raita, joka sisältää ”musiikkivideon”. Ohjelman katselu ei tosin ole helppoa, sillä Spectrumissa ei ole LP-asemaa. Levysoittimen voi kytkeä kasettiaseman tilalle koska liitin on standardinmukainen, mutta *XL1*:n ohje kehottaa kopioimaan ohjelman kasetille ”oikealla nopeudella ja voimakkuudella” (joita ei määritellä).

XL1:n tekijöiden ajatus oli, että ohjelma ladataan visuaaleiksi, kun LP-levyä kuunnellaan. Mutta eikö sen voisi ladata suoraan LP:ltä? Kokeillaan emulaattorilla. Hain autenttiset vinyylisoittimet isovanhempien varastosta: komea vahvistimella varustettu B&O Beocenter 3500 70-luvulta (kuvassa) ja vähemmän komea vahvistamaton Philips FP140A 80-luvulta.

Kerroimme Skrollissa 2018.4 ZX Spectrumin kasettiohjelmien siirtämisestä nykykoneelle: kytketään soitin tietokoneen mikrofonituloon, nauhoitetaan vaikkapa Audacityllä (s. 24, audacityteam.org) WAV-tiedosto ja muunnetaan se ZX Spectrum -muotoon MakeTZX-ohjelmalla (ramsoft.bbk.org.omegahg.com/maketzx.html). Lähdin tästä liikkeelle.

Äänen lähdöstä ja tulosta on kytkettävä äänenkäsittelyt pois, ja voimakkuuden on oltava sopiva. Tarvitaan myös kaapelit. Käytin Røden SC7 TRS-TRRS-sovitinta, jolla voi kytkeä kolminapaisella 3,5 mm:n jakkiliittimellä (*miniplug*) varustetun äänilähteen tietokoneen nelinapaisen äänijakin mikrofonipuoleen.

Philipsin soittimessa oli RCA-lähdöt, joihin kytkin jakkiadapterit ja liitin ne Røden jatkeena PC:hen. B&O:ta kokeilin sekä vahvistamattomasta RCA-lähdöstä että 3,5-6,3 mm -adapterilla vahvistetusta kuulokelähdöstä. Vanhan koulun B&O:ssa oli myös vahvistamaton DIN-liitäntä, sekin olisi vaihtoehto DIN-RCA-sovittimella.

Phonon probleema

Yksi ongelma levysoittimelta ”lataamisessa” on äänisignaalin taso. Se on vaimea ja edellyttää lisäksi usein RIAA-taajuuskorjausta (Audacityssa *Effect > Equalization > Curve: RIAA*). Tavallisesti vahvistin huolehtii näistä Phono-liitännällään.

Muitakin vaaranpaikkoja paljastui: Philips särki ääntä ja hyppyytti neulaa, mitä ei datasuhinasta heti huomannut. B&O ei särkenyt, ja sen vahvistimelta sai korjatun signaalin, mutta se pyöritti levyä liian hitaasti ja pudotti moottorinsa hihnan. Philipsin rahinat eivät ratkenneet neulaharjalla, mutta uusi neula auttoi. B&O:n hihnan sai paikalleen ja nopeutta säädettyä Vernier-rullasta (33⅓ RPM), mutta uusi hihna ja moottorin verryttely olivat lopulta tarpeen.

Korjausten jälkeen B&O:n vahvistamaton ja vahvistettu lähtö (volyymi esim. 20 % ja mikrofoni 30 %) tuottivat LP:ltä TZX-tiedoston emulaattorikäyttöön. Mutta onnistuisiko se suoralataus? Käynnistin emulaattorin ja mikrofonikasettilatauksen. Vinyyli pyörimään… Spectaculator, SpecEmu, ZXSpin (Windows) sekä Zxsp (Mac) löysivät bittivirran, mutta lataus ei mennyt läpi.

Philips oli hyödytön ilman vahvistinta, mutta vahvistimen kautta kytkettynä tulos oli sama: toimiva TZX-tiedosto syntyi helposti, mutta suora lataus ei toimi. Kävin läpi 3 tietokonetta, 4 emulaattoria, 3–4 eri kytkentätapaa ja 2 levysoitinta samalla tuloksella. Kokeilin tuloksetta myös yksittäisiä stereokanavia, ylä- ja alasuodattimia sekä äänen eri tasoja ja tasokorjauksia.

Suora lataus löytää ohjelman nimen (aloituskuvassa), mutta katkeaa aina, kun varsinainen datakohina alkaa. Suurin ongelma onkin, että LP-lähteen ääni on liian hyvä. Taajuusvaste poikkeaa liikaa halvan kasettimankan ininästä. Kopiointi kasetille huonontaisi ääntä sopivasti.

Ei ihme, ettei ”LP-ROMista” tullut juttua. Mutta onpa komea levyasema!

P.S. Vuosina 1983–1984 Thompson Twins teki samantapaisen vinyylitempun ja Shakin’ Stevens sekä The Stranglers julkaisivat dataa musiikkikaseteillaan. Vuonna 1989 Codemasters julkaisi levyllisen softaa cd:llä – siis ääni-cd:llä, ei cd-romilla. Pyhä modeemi, Batman.

Katso XL1:n ”musiikkivideo” YouTubessa, hakusanat: XL1 Spectrum. Kyseessä on varhainen demo, vieläpä synkatuilla lyriikoilla.

Pelin ympärille syntyi hylkiöiden kultti

Mies, joka pelasti Harvesterin

Aarno Malin bloggaa huumeista ja kirjoittaa freelancer-toimittajana tutkivaa journalismia yhteiskunnan laitamilta. Tietokonekulttuurin maailmassa hänet tunnetaan Harvester-seikkailupelin johtavana asiantuntijana. YouTubessa julkaistavan The Harvester Show'n myötä myös Malin itse on saanut oman fanikuntansa.

Teksti: Miika Auvinen
Kuvat: Nanne Kukkura, Miika Auvinen

Vuonna 1996 julkaistu *Harvester* oli useita vuosia unohdettu kauhuseikkailupeli. Yksi maailman pahamaineisimmista peleistä sisältää kyseenalaista sisältöä kannibalismista hämäriin seksuaalissävytteisiin kohtauksiin. Teosta pidetään eräänlaisena häiriintyneiden videopelien kruununjalokivenä.

Se, että peli on löytänyt tiensä omituisuuksien ystävien kokoelmiin tänäkin päivänä, on osittain **Aarno Malinin** ansiota. Hänen Facebookissa ylläpitämänsä *Harvester*-fanisivu ja YouTubessa ilmestyvä *The Harvester Show* antoivat pelille huomiota ja nostivat sen takaisin seikkailu- ja kauhupelien ystävien tajuntaan. Malinilla on myös oma roolinsa pelin tuomisessa nyky-yleisön pelattavaksi: hän toimi asiantuntija-apuna, kun Good Old Games -verkkokauppa otti teoksen kokoelmiinsa. Esimerkiksi **Juho Kuorikosken** teoksessa *Suuret seikkailupelit – tietokonepelien klassikot* Malinia kutsutaan maailman johtavaksi *Harvester*-asiantuntijaksi.

Sen lisäksi, että Malin ylläpitää *Harvesterin* Facebook-fanisivua ja peliin keskittyvää YouTube-sarjaa, on hänestä itsestäänkin tullut fanituksen kohde. Miehen karisma tuntuu purevan katsojiin, sillä YouTuben kommenteista löytyy paljon häneen itseensä kohdistuvaa hehkutusta.

Malinin tie pelin ylimmäksi mestariksi oli ohdakkeinen: hänen menneisyytensä oli masennuksen ja terveysongelmien kyllästämä. Pelin fanitus, sen tekijöiden haastattelut ja pelimateriaalin kerääminen olivat kuitenkin kuin pieni seikkailu, jotka auttoivat Malinia vaikeina aikoina.

Harvester pelasti elämän

Aarno Malin kertoo sairastelleensa nuorena. Hänen jalkansa tulivat kipeiksi, jolloin kävely muuttui hitaaksi nilkutteluksi. Kukaan ei tuntunut uskovan nuorta kivuista kärsivää miestä, ja hän joutui lopulta jättämään lukio-opintonsa kesken. Kipeät jalat ja se, etteivät edes lääkärit uskoneet häntä, johti masennukseen, ahdistukseen ja unettomuuteen. Pahimmillaan Malin joutui nuorisopsykiatrian poliklinikalle.

Eräissä allergiatesteissä löytyi sattumalta syy kipuihin: keuhkosarkoidoosi. Malinin terveydentila oli kuitenkin jo niin huono, että hän eristäytyi kotiinsa. Ajan myötä ilmeni muita kipuja, joita ei myöskään uskottu. Näiden kipujen syiksi todettiin myöhemmin selkärankareuma.

Pahimmillaan Malin viis veisasi elämästään. Hän tutustui myrkkyihin: keräsi, kuivasi ja söi muun muassa ukonhatun juuria, jotka sisältävät tappavaa akonitiinia.

- Tuossa jamassa olin tutustuessani *Harvesteriin*, Malin toteaa.

Hän jäi vuonna 2015 työkyvyttömyyseläkkeelle. Kipujen vuoksi hän joutuu viettämään paljon aikaansa kotona. Toisaalta tämä mahdollistaa show'n tekemisen.

Nykyään Malin voi jo paremmin. Lapsen saaminen kannusti Malinia tutkailemaan menneisyyttään tarkemmin, ja hän haluaa tulla sinuiksi men-

neiden vaikeampien aikojen kanssa. Freelance-toimittajan töillä Malin saa jonkin verran ylimääräisiä tuloja. Auto mahdollistaa liikkumisen ulkona. Uutena ilmiönä on tullut erilaisten suurten medioiden mielenkiinto Malinia ja tämän työtä kohtaan.

Voi olla, että *Harvester* pelasti Malinin elämän.

- Jollain selittämättömällä tavalla *Harvesterin* maailmaan uppoutuminen pahimpina hetkinä toi oudon kotiinpaluun olon, ja siinä maailmassa oli hyvä olla. Tämän vuoksi jäin peliin harhailemaan ja juttelemaan ihmisten kanssa. Peli herätti valtavan skaalan erilaisia tunteita. Inhon ja pelon lisäksi esimerkiksi ihastuin erääseen pelihahmoon.

Pelkkä pelaaminen ei riittänyt. Malin halusi kaivautua syvemmälle pelin maailmaan, ja alkoi jäljittämään tekijöitä ja peliin liittyviä materiaaleja.

Salapoliisityötä ja internet-seikkailua

Harvesterin ilmestyessä jo pelkkä sensuroimattoman version saaminen oli hankalaa. Malin joutui soittamaan lukuisia pelikauppoja läpi löytääkseen oikean, yhdysvaltalaisen version. Eurooppalaisesta versiosta puuttui kohtaus, jossa lapset söivät äitinsä elävältä. Kun sensuroimaton versio löytyi, peli vei sydämen.

Malin halusi tietää kaiken *Harvesterista*. Pelin tekijät olivat hajaantuneet ympäriinsä ja käyttivät eri nimiä kuin peliä tehdessään. Yhteyden saaminen oli hankalaa ja vaati kaikkien mahdollisten hakusanojen käyttämistä – sekä välillä silkkaa tuuria.

Esimerkiksi pelin naispääosan ja Malinin peli-ihastuksen Stephanien esittäjä **Lisa Cangelosi** löytyi lopulta tuurilla Facebookista. Ensimmäinen yhteydenpitojakso tyrehtyi. Malin pohtii, vaikuttivatko tyrehtymiseen hänen fanisivujensa materiaalit, joissa Lisa esiintyi vähäisissä vaatteissa: Cangelosista kun oli tullut uudelleensyntynyt kristitty. Lopulta yhteydenpito kuitenkin jatkui. Cangelosin haastattelu oli kolmen vuoden prosessi, ja varsinainen teksti oli koottu tilkkutäkkimäisestä aineistosta tältä ajanjaksolta.

Miksi pelin päähenkilö Steven äiti leipoo päivittäin keksejä, mutta heittää ne kaikki roskikseen?

Ajan myötä Malin tutustui myös pelin muihin näyttelijöihin, ohjelmoijiin ja esimerkiksi pelin ohjaaja-käsikirjoittaja **Gilbert Austiniin**. Tekijät ovat lähettäneet Malinille materiaaleja, kuten kulissien takaisia valokuvia, vanhoja lehtijuttuja ja promootiomateriaalia.

Harvester-fanisivut ja The Harvester Show

Facebookin *Harvester*-fanisivuilta löytyy esimerkiksi pelin arvosteluja ja mainoksia. Esimerkiksi puolalaiset fanit olivat innoissaan Malinin julkaistua puolalaisia arvosteluja ja ennakoita pelistä. Eräs innokas fani käänsi puolankieliset tekstit englanniksi.

Sivuilla on myös manuaaleja ja alkuperäisiä luonnoksia pelin hirviöistä. Kenties parasta antia ovat lukuisat haastattelut.

Aluksi Malin ylläpiti Facebookin *Harvester*-sivua nimettömänä. Jossain vaiheessa, esimerkiksi haastatteluja julkistaessaan, hänen oli kuitenkin pakko tulla esille omalla nimellään.

Malin julkaisee YouTubessa Lodge Level 4 -kanavallaan *Harvesterille* omistettua sarjaa *The Harvester Show*. Kanavalla on tällä hetkellä yli 1 200 tilaajaa.

Show´ssa käydään läpi kulissien takaisia ilmiöitä näyttelijöistä aina pelin hämärimpiin yksityiskohtiin asti. Miksi pelin päähenkilö Steven äiti leipoo päivittäin keksejä, mutta heittää ne kaikki roskikseen? Entä miksi siteissä sängyssä makaava Steven isä on lukittu makuuhuoneeseen siten, että nähdäkseen isänsä Steven on kiivettävä huoneeseen ikkunan kautta? Malin kertoo rauhallisella äänellä pelin häiriintyneistä yksityiskohdista ja raottaa pelin pinnan alla piileviä salaisuuksia. Synkin jakso kertoo erään pelin näyttelijän lapsipornotuomiosta ja Malinin moraalisesta kamppailusta sen suhteen, haluaako hän käsitellä aihetta julkisesti. Uusimmilla videoilla Malin kertoo matkastaan *Harvester*-asiantuntijaksi ja myös siitä, miten hän jäljitti pelin tekijöitä – usein vuosikausien ajan. Sarja on hyvin viihdyttävä myös heille, jotka eivät ole pelin varsinaisia faneja.

- Vaikka et ole *Harvester*-fani, voit saada *The Harvester Show´sta* paljonkin irti. *Harvester Show* kertoo peliteollisuudesta ja aikakaudesta yleensäkin, Malin kertoo.

Malin myös pelaa *Harvesteria* YouTubessa yhdessä ystävänsä **Stephenin** kanssa. Stephen muokkaa pelivideot, ja hän esimerkiksi editoi vitsejä videoiden sekaan.

- Teemme pelivideot siten, että näymme molemmat ruudulla ja jutustelemme samalla kun pelaamme. Stephen on positiivisia *Harvester*-tuttuja. Hän työskentelee New Yorkissa firmassa, joka tekee videoita, joten

hän tuntee tekniikan ja teknologian hyvin.

Eräs kanadalainen fani alkoi tuottamaan *The Harvester Show* -tuotteita, sillä hän fanittaa itse pelin lisäksi myös Malinia. Fanituotteisiin kuuluvat esimerkiksi paidat ja mukit. *Harvesterin* ja *The Harvester Show´n* faneista muodostuu yhteisö, johon kuuluu tavallisia ihmisiä, mutta myös jos jonkinmoisia outoja hylkiöitä.

Faniyhteisö kasvaa

Malin kertoo tutustuneensa *Harvesterin* kautta moniin uusiin ystäviin. Hän on ollut yhteydessä *Harvesterin* tekijöihin, ja parhaimmillaan kanssafanien kanssa on syntynyt syviäkin ystävyyssuhteita. Malin halusi alusta saakka luoda yhteisön. Hän esimerkiksi aluksi henkilökohtaisesti toivotti tervetulleeksi Facebook-sivusta tykänneet. Nykyään tykkääjiä on niin paljon, että kaikkiin ei voi enää ottaa henkilökohtaisesti yhteyttä.

Harvesterin fanisivujen ympärille on syntynyt yhteisö, jonka jäsenet ovat Malinin kanssa joskus tiiviissäkin yhteydessä. Miittejä ei ole vielä ollut, sillä yhteisö on jakaantunut ympäri maapalloa. Faneja löytyy esimerkiksi Australiasta, Brasiliasta ja eri puolilta Eurooppaa. Malin on kuitenkin haaveillut miitistä, jonka voisi pitää jonkun suuremman pelitapahtuman kupeessa.

Pohdimme, kuinka häiriintynyt peli kerää yhteen tietynlaisia ihmisiä.

- Jo faniyhteisöä perustaessani aavistin, että monilla faneilla voisi olla ongelmia elämässään. Visioni oli, että sivustoni voisi yhdistää heitä kaikkia. Kun olemme asian äärellä, unohdamme riidat ja toistemme haukkumiset, ja keskitymme fanittamaan *Harvesteria*.

Joskus ongelmat ovat astetta syvempiä. Mukavien ihmisten joukossa on Malinin mukaan myös täysin seinähulluja tapauksia. Malin kertoo, että on onnellinen, etteivät pahimmat tapaukset asu Suomessa, sillä hän pelkäisi heitä.

- Jotkut luulevat *Harvesterin* lähettävän heille viestejä, ja että peliin olisi koodattu salaista viisautta ja salaisia viestejä. Kun olet tarpeeksi korkealla tasolla, eli tarpeeksi sekaisin, tajuat viestit. Ei hyvänen aika.

- Jotkut luulevat Harvesterin lähettävän heille viestejä, ja että peliin olisi koodattu salaista viisautta ja salaisia viestejä

Malin kertoo, että hänellä on hyvät välit pelin suunnittelijoihin ja näyttelijöihin. Esimerkiksi **Mike Napodano**, eräs pelin pääsuunnittelijoista, lähetti Malinille laatikollisen tavaraa, kuten alkuperäisen käsikirjoituksen ja roolituspalvelun dokumentin, jonka perusteella näyttelijät oli valittu peliin. Tätä materiaalia Malin on käyttänyt esimerkiksi *The Harvester Show'n* pohjana.

Unelmissa Harvester-dokumenttielokuva

Malinilla on monia intohimoja: hän kirjoittaa freelancer-toimittajana juttuja Suomen Kuvalehteen ja ylläpitää huumausaineisiin ja niiden kauppaan keskittyvää Punainen kolmio -blogia. Tämän jutun kirjoittamisen aikana Yle Puheella oli Malinin haastattelu hänen huumekauppaa käsittelevän tutkivan journalisminsa vuoksi. Viime kuukausina Malinin nimeen on saattanut törmätä paljon somehuomiota saaneen artikkelin myötä, jonka vuoksi myös MTV3 julkaisi Malinista huhtikuussa jutun. Juttu oli otsikoitu seuraavasti:

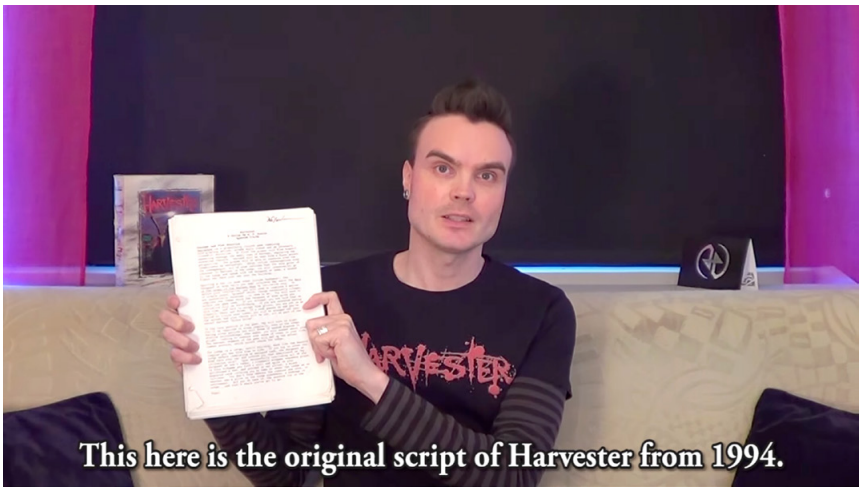
Aarno Malin kahlasi lastenkidutuspornoa lähes kahden vuoden ajan kirjoittaessaan kammottavista pedofiiliyhteisöistä – kunnes yhtäkkiä tajusi tarvitsevansa kipeästi ammattiapua: "Sairaanhoitajatkin järkyttyivät, istuimme vain hiljaa"

Tutkimustyö artikkelia varten vei Malinin internetin pimeimpiin kolkkiin.

Tutkimustyö artikkelia varten vei Malinin internetin pimeimpiin kolkkiin. Malin kertoo, ettei enää voi esimerkiksi suhtautua väkivaltaviihteeseen samalla tavalla: kun on nähnyt, miten raakaa tosielämä pahimmillaan on, menettävät fiktiiviset hirmutyöt tehoaan. *Harvesteria* Malin ei silti hylännyt – pelin juoni on ovela ja saa pelaajan itsensä kyseenalaistamaan omat tekonsa pelin sisällä. Kyseessä ei ole geneerinen verellä mässäily, vaan pelaajan halutaan haastavan suhdettaan peliväkivaltaan.

Malin toivoo, että pystyisi tekemään tekijöistä videohaastatteluita. Suurena haaveena Malin unelmoi matkaamisesta Teksasiin ja dokumenttielokuvan tekemisestä *Harvesterista*, sen tekijöistä ja syntyprosessista. Tämä vaatisi rahoitusta, kunnon kameran ja lentoliput Teksasiin. Lisäksi on materiaaleja, joita Malin toivoo pääsevänsä joskus käyttämään. Hän on kuullut VHS-kaseteista, joille on tallennettu pelin tekoprosessia.

- Monet tekijät ovat jo toivottaneet minut tervetulleeksi Teksasiin, mutta minun kunnollani ei noin vain lähdetäkään Yhdysvaltoihin.



Aarno Malin ja Harvesterin käsikirjoitus The Harvester Show'n jaksossa 5.

Overwatch-turnauksen alkuvalmistelut Assembly Summer 2018 -tapahtumassa.

E-urheilun suomenmestaruus 2019 – järjestelyjen takana

Suomen elektronisen urheilun liitto SEUL ry:n järjestämien e-urheilukisojen SM-vuosi 2019 käynnistyi huhtikuun lopulla. Eri pelien karsintoja on järjestetty yhteistyössä operaattoreiden kanssa kevään edetessä. Miten lajiliitto vaikuttaa SM-kisojen järjestämiseen? Entä mikä on operaattoreiden rooli?

Teksti: Janne Laukkanen, Tomi Pyy
Kuvat: SEUL ry / Joel Tainio

E-urheilun SM-kisavuosi saatettiin liikkeelle huhtikuun lopulla *Super Smash Bros.*- ja *Overwatch*-pelien karsinnoilla. Kisoja järjestää Suomen elektronisen urheilun liitto eli SEUL ry yhdessä eri operaattoreiden eli järjestävien tahojen kanssa. Operaattoreiden joukkoon kuuluvat esimerkiksi Porin Insomnia-tapahtumia järjestävä Insomnia ry ja Assembly Organizationin hiljattain korvannut Elisa Viihde.

SEUL ry:n verkkosivujen mukaan suomenmestaruudesta kisaillaan tänä vuonna seitsemässätoista pelissä, ja järjestelyissä on mukana kaksitoista eri operaattoria. Lajin isojen nimien kuten *Counter-Strike: Global Offensiven*, *Overwatchin* ja *Dota 2:n* lisäksi listoille kuuluvat esimerkiksi virtuaalikorttipeli *Hearthstone*, *Tekken 7* ja useampi mobiilipeli. Ensimmäiset karsinnat aloitettiin huhtikuussa, ja kisat huipentuvat alkusyksyllä järjestettäviin finaaleihin.

Mikä ihmeen SEUL, ja mikä operaattori?

SEUL ry on vuonna 2010 perustettu suomalaisen kilpapelaamisen keskusjärjestö. Liiton tavoitteena on kehittää elektronista urheilua ja nostaa lajin tunnettavuutta sekä arvostettavuutta harrastuksena. Lajiliiton tehtäviin kuuluu myös kansallisten mestaruuskisojen toimeenpano e-urheilun eri lajeissa. E-urheilun SM-turnaustoiminnasta vastaa siis SEUL ry:n hallitus, joka nimittää yhden tai useamman projektipäällikön. Projektipäälliköt valvovat, että kisat järjestetään liiton edellyttämällä tavalla. SEUL ry on ulkoistanut turnausten ja pelisarjojen käytännön järjestelyt ja hallinnon pitkälti kolmansille osapuolille, jotka tunnetaan operaattoreina. Liitto itse siis valvoo kisojen järjestelyjen sujuvan vaatimusten mukaisesti sekä hoitaa esimerkiksi kurinpitovaliokuntaa ja pelaajarekisteriä ja toimii kisojen eikaupallisena osapuolena. Liiton piiriin kuuluu myös tuomaritoiminnan järjestäminen ja tukeminen yhdessä operaattorin kanssa, mahdollinen rahallinen tuki tiettyihin kilpailuun liittyviin kuluihin sekä mitalien ja pokaalien hankkiminen turnausten käyttöön.

Kurinpidon puolesta SEUL ry vastaa eSM-turnauksissa myös antidoping- ja tuomaritoiminnasta. Kurinpitovaliokuntaa voidaan tarvita esimerkiksi tilanteissa, joissa eSM-turnaukseen osallistuva pelaaja ei saa tasavertaista kohtelua turnauksen aikana. Valiokunnan olemassaoloa ja käyttöä liitto perustelee uskottavuudella.

”SEULin näkemyksen mukaan kaupalliset toimijat eivät voi olla itse päättämässä ratkaisuista ollessaan kiistan osapuolena”, kertoo SEUL ry:n puheenjohtaja **Joonas Kapiainen.**

”Liitto toimii muutoinkin hyvin samankaltaisesti kuin perinteisten urheilulajien lajiliitot.”

Operaattorit ovat tahoja, joille SEUL ry on myöntänyt eSM- eli e-urheilun

SM-kisojen järjestämisoikeuden. Operaattoreiden tehtäviin kuuluu tapahtumien kaupallisen toiminnan organisointi. Ne vastaavat esimerkiksi online- ja offline-turnauksista, turnauksissa käytettävästä laitteistosta, pelien suoratoistosta ja tapahtumien markkinoinnista. Esimerkiksi vuoden 2019 *Counter-Strike: Global Offensiven* operaattorina toimii Elisa Viihde ja *Dota 2:ssa* Vectorama. Vaikka operaattorit vastaavatkin kisojen järjestelyistä pääosin itsenäisesti, SEUL ry tekee tahojen kanssa tiivistä yhteistyötä. Operaattorit esimerkiksi sitoutuvat kilpailutuksen yhteydessä liiton laatimiin ohjesääntöihin, joilla varmistetaan kisojen laatu.

”Tapahtumaa järjestävän operaattorin on suostuttava laadullisiin standardeihin ja arvosteluperusteisiin. E-urheilun SM-kisat ovat SEUL ry:n toimeenpanemia tapahtumia, joten liitolla on todellinen mahdollisuus ottaa kantaa ja vaikuttaa”, Kapiainen selventää.

Muutoksia tulevalle kisavuodelle - kansainvälisyys kehiin

SEUL ry kilpailutti alkukeväästä operaattoreita järjestämään eri pelien eSM-kisoja ja valitsi kilpailutuksen perusteella kuusi uutta järjestävää tahoa. Yksi näistä on Elisa Viihde, joka on ensimmäistä kertaa järjestämässä *CS:GO:n* eSM-kisoja. Aiempina vuosina Valven ikonisen räiskintäpelin SM-kisoja on hoitanut Assembly Organizing Oy, joka keskittyy jatkossa Telia E-Sports Series -kisoihin. Elisa Viihde ei kuitenkaan ole ensikertalainen e-urheilutuotannossa. Viihdepalvelun kehityspäällikkö **Teemu Koski** toteaa, että kokemusta lajin parista löytyy vuodesta 2017 alkaen.

ENCE:n tähtipelaaja Aleksi ”allu” Jalli.

”Ensimmäistä kertaa taisimme järjestää oman e-urheilutuotantomme Porin Insomnia-tapahtumassa tuolloin 2017. Lähdimme liikkeelle Hearthstonesta, ja seuraavana vuonna näytimme LanTrek-tapahtuman kaikki pelit palvelussamme. *CS:GO*:ta näytimme LanTrekin lisäksi Vectoramasta.”

Koski kertoo, että Elisan tavoitteena on erottua aiemmasta kisajärjestäjästä esimerkiksi laadukkaalla tuotannolla ja kansainvälisellä kulmalla. Kansainvälisyyttä pyritään korostamaan Elisan kisatoiminnassa yhdistämällä *CS:GO:n* SM-kisat Arctic Invitational -kutsuvierasturnaukseen. Arctic Invitational on suomalaisen Starsquad Oy:n ja Elisan järjestämä turnaus, johon kutsutaan kolme kansainvälisesti menestynyttä joukkuetta ja suomalainen kutsujoukkue. Ensimmäinen julkistettu joukkue on yhdysvaltalainen NRG Esports. Neljäs paikka on varattu eSM-kisojen voittajille, jotka kutsutaan kamppailemaan kansainvälisten joukkueiden kanssa 100 000 euron palkintopotista Hartwall Arenalle syyskuussa. Pääosassa turnauksessa on *CS:GO*, mutta tapahtumassa järjestetään myös Elisan, OnePlus-älypuhelinvalmistajan ja Hatchet Entertainmentin mobiilipelikiertueen finaali.

Tuotannossaan Elisa Viihde aikoo panostaa esimerkiksi kuvanlaatuun ja laaja-alaiseen saatavuuteen. Kehityspäällikkö Kosken mukaan Elisa Viihde aikoo näyttää *CS:GO*-kisoja perinteisten stream- eli suoratoistopalveluiden lisäksi myös omilla kanavillaan.

”Haluamme tarjota mahdollisimman kattavat mahdollisuudet seurata e-urheilua. Elisa tulee tarjoamaan kisan suoratoistolähetyksen Viihde-palvelun lisäksi kaapeli-TV:ssä, Elisa Sport -palvelussa sekä myös Twitch TV -suoratoistopalvelussa, johon kaikki sisältö tulee ilmaiseksi sellaisenaan. Maksullisen palvelumme kautta lähetyksiin on mahdollista tarjota 4k-kuvanlaatua, jota Twitch ei tällä hetkellä mahdollista”, Koski avaa.

Kosken mukaan Elisa Viihde on lähtenyt mukaan e-urheiluun kehittääkseen omaa palveluaan. Kehityspäällikkö pitää kuitenkin tärkeänä, että he voivat toimia yhdessä SEUL ry:n kanssa e-urheilun kehittämiseksi ja tukemiseksi Suomessa.

”Elisa Viihteen osalta voin todeta, että haluamme tuoda viihdettä, hyvää urheilusisältöä ja laajentaa lajivalikoimaamme entisestään. Tämä on kasvava laji ja tärkeä harrastus monelle suomalaiselle. Haluamme tietenkin olla mukana tukemassa ja kehittämässä harrastusta ja siihen liittyvää urheilutoimintaa”.

Muutosten mahdolliset vaikutukset

Vuosi 2019 on kiinnostava eSM-urheilun kannalta. Edellisvuoteen verrattuna kisailtavien pelien määrä on kasvanut kymmenestä kuuteentoista, ja uusina pelityyppeinä nähdään esimerkiksi tanssipelejä ja *Smash Bros.* -sarjan pelejä. Assembly Organizingin korvautuminen Elisa Viihteellä tarjoaa suomalaispelaajille uusia mahdollisuuksia lyödä itsensä läpi myös kansainvälisillä pelikentillä, mutta tilanne aiheuttaa harrastukselle myös varjopuolensa. E-urheilukisojen lähetysten siirtyminen yksityisten viihdepalveluiden piiriin tuo mukanaan riskin muuttaa e-urheilun seuraamisen suuntaa poispäin valtakulttuurista.

SEUL ry:n eSM-säädöksissä vaaditaan, että kisoista on järjestettävä ilmainen suomenkielinen lähetys, joka on helppo toteuttaa esimerkiksi Twitch TV:n kautta. Monelle tietokoneharrastajalle Twitch-suoratoiston seuraaminen ei ole ongelma, mutta mikäli kisoja ei tulevaisuudessa näytetä esimerkiksi enää Yleisradion kanavilla, moni ainoastaan television kautta lajia seuraava voi jäädä kyydistä. Lisäksi tilanne voi vähentää uusien katsojien määrää. Mikäli televisiolähetysten siirtäminen maksumuurien taakse osoittautuu kannattavaksi, on riski, ettei Yle pysy enää mukana. Tilanne korostuu esimerkiksi Assembly Organizingin vetäydyttyä eSM-järjestäjyydestään Telia E-Sports Seriesin pariin: Assembly ei ole näin enää sidottu SEUL:in eSM-järjestysoikeuden vaatimukseen ilmaisesta lähetyksestä. On todennäköistä, että Yle tulee muutoksista huolimatta jatkossakin näyttämään kanavillaan kansainvälistä e-urheilua.

Sekä SEUL ry että Elisa Viihde ovat kuitenkin kertoneet pyrkivänsä edistämään e-urheilua sekä lajina että harrastuksena Suomessa, eikä isojen toimijoiden tarjoamia uusia mahdollisuuksia voi kiistää.

Onko Saabissa Pong?

Sitkeä huhu väittää, että Saabin moottorinhallinta-yksikköön olisi piilotettu Pong-peli. Skrolli tutkii.

Teksti ja kuvat: Janne Sirén

Tesla lisäsi hiljattain autoihinsa Atari-pelejä ”pääsiäismunina” (*easter egg*, Skrollit 2018.4, 2019.1). Pelit ovat todellisia, vaikkakin katoavaisia (s. 80). Huhun mukaan tämä ei kuitenkaan ollut ensimmäinen kerta, kun autonvalmistaja on piilottanut easter egg -pelin autoonsa.

Tarinan mukaan nimittäin Saabin 9-3-mallissa on Pong, johon pääsee käsiksi Tech 2 -huoltoskannerilla. Kerroimme Pongin historiasta viime Skrollissa – onko tässä jatkoa sille vaiko saman lehden Illuminatus-puijausjutulle?

”Saab Pong”

Haku löytää internetistä paljon osumia, jopa kuvia. Mutta kun katsoo tarkemmin, huomaa, että sosiaalisen median fiilistelyt ovat johdannaisia tästä viestistä: ”When using the GM Tech II on a Saab 9-3 SS (I believe this works on 2003–2007 cars) if you go into the Airbag/SRS system submenu, there is an easter egg that allows the tech to play Pong on the Tech II.”

Mitä ilmeisimmin viestin on lähettänyt nimimerkki **Highball** Jalopnik-verkkolehteen. Jalopnik on julkaissut sen alkuvuodesta 2013. Mukana on vinkkaajan lähettämä valokuva, jossa näkyy osa Tech 2:n käynnistysruudusta.

Googlen kuvahaku ei löydä vanhempia otoksia kuvasta, joten se voi olla Highballin ottama. Toisaalta jos vinkkaat lehdelle Pong-pelistä ja otat valokuvankin, miksi et ottaisi kuvaa itse pelistä? Sosiaalisesta mediasta löytyneet ”lisäkuvat” osoittautuivat geneerisiksi Pong-grafiikoiksi. Moni on kysellyt vahvistusta Saab-huhulle sitä saamatta.

Saabin sekä muiden pienten ja lopettaneiden yritysten kohdalla luotettava tieto voi olla kiven alla. Kuten huomasimme tarkistaessamme Salora Fellow’n muistimäärää Skrolliin 2018.3 (ks. skrolli.fi/blogi), vaarana on, että tietolähteet toistavat samaa väärää tietoa. Salora Fellowssa oli yleisesti uskottua vähemmän muistia, joten nyt aloimme epäillä, oliko Saabissa Pongia lainkaan.

Aprillia?

Seuraava löydös ei kasvattanut luottamusta. Vuonna 2011 nimimerkki **quinophex** kirjoitti Saab Scene -foorumille[1]: ”Näin Tech 2:n 9-3:n (2005) Airbag-diagnostiikkavalikossa Pong-vaihtoehdon. Painoin nappia valmistautuen kenties turvatyynyjen laukeamiseen. Sen sijaan löysinkin tervetulotoivotuksen Tech 2:n Pongiin ja täyden Pong-version, jota pystyi pelaamaan.”

Itse vinkissä ei ole vikaa, mutta sen päiväys on 31.3.2011. Aikaerosta riippuen viesti on voitu kirjoittaa aprillipäivänä. Vastaukset ovat vitsailevia, joskin **tonyfisk** lausuu tahattoman profetiaalisesti: ”Ehkäpä jossain vaiheessa peli ilmestyy auton näytölle ja mailoja ohjataan ratin painikkeilla” – Teslan Atari-pelit toimivat tähän tapaan.

Varhaisin Google-maininta Saabin Pongista löytyi SaabCentral-foorumilta vuodelta 2008. Nimimerkki **turboAERO** kirjoitti autonsa Tech 2 -käsittelystä: ”Pingin sijaan siellä oli Pong-nappi, jota painettaessa Pong-peli ohjelmoitiin ECUuun. Pystyit pelaamaan sitä Tech 2:lla ja sitten poistumaan pelistä siirtyäksesi ping-tilaan.”

Raflaavimpana väitteenä turboAERO esitti, että Pong-peli olisi ohjelmoitu nimenomaan ECUun eli auton moottorinhallintayksikköön. TurboAERO jopa kysyy: ”Tarkistakaa, koskeeko tämä vain 9-3 (2002) Linearia vai laittoivatko he Pongin kaikkiin T7-tietokoneisiin?” – T7 eli Trionic 7 on Saab 9-3:ssa käytetty moottorinhallintayksikkö.

Ketju sai pari lyhyttä vastausta. **Ubermich** komppasi lisätiedolla: ”Tiedän, että tämä toimii kaikissa vuodesta 1999 ylöspäin, paitsi epä-Saabeissa 9-7X ja 9-2X.” **Tweek’s Turbos** vastasi ”Jep” ja linkkasi SaabLink-foorumille. Linkki ei kuitenkaan auennut ilman foorumitunnuksia. Koska Google ja Bing oli nyt kaluttu, jäimme odottelemaan pääsyä SaabLinkiin.

Ping pong

TurboAEROn viesti sisälsi kyllä uskottavalta kuulostavan selityksen sille, miksi Saabissa voisi olla Pong: sanaleikki ping pong. Saabin omistaneen General Motorsin huoltoskannereissa auton elektroniikkaa pystyi ”pingaamaan”, kuin tietokoneita verkossa konsanaan. Saab ei tosin käyttänyt ping-termiä omissa skanneriversioissaan.

Toinen mahdollinen etymologia löytyy sieltä moottorinhallintayksikön suunnalta: moottorin pingaus eli nakutus. Kolmas selitys voisi olla, kuten

[1] *Artikkelin sitaatteja on suomennettu englannista ja tiivistetty.*

quinophex vitsaili, että poksahtamiseen viittaava ”Pong” on hassunhauska juttu turvatyynyvalikossa.

Ei siis ole mahdotonta, että Saabin ohjelmistokehittäjät olisivat keksineet lisätä Pongin skanneriin. Allekirjoittanut on syyllistynyt vastaavaan itsekin: tehdessäni Offline Orbit -etälukijaa 90-luvulla (s. 20), käytin kohtuuttomasti aikaa lisätäkseni siihen numerolaattapelin. Microsoft pisti vielä paremmaksi: Excelistä löytyi jopa lentosimulaattori.

Toisaalta väite siitä, että Pong-peli olisi Saabin moottorinhallintayksikössä, lienee silkkaa tietämättömyyttä. Piilopeliä tuskin ujutettaisiin niin kriittiseen paikkaan. Tokkopa se on muuallakaan autossa, kun tarkastelee Tech 2 -skannerin tekniikkaa.

Tech 2

Autohuoltojen lisäksi Tech 2 -skanneria löytyy erilaisina versiona ihan myynnistä. Tarvittavat TIS-ohjelmistot voi edelleen lisensoida kuukausitilauksena. Tarjolla on jopa Tech2Win-ohjelmaemulaattori. Tuotteiden laatu tosin vaihtelee suuresti.

Kaikissa Tech 2:sissa on kuitenkin sama ”vika”: yhteydet ulkomaailmaan ovat sarjaliikennetasoisia ja niiden 1990-luvun tiedonsiirtotekniikka on hidasta. Niinpä skannereiden ohjelmistoa tai valikoita ei ladata autosta, vaan laitteen sisältä luottokorttipinkan kokoiselta 32 megatavun (!) PCMCIA-muistikortilta. Tiedon lataaminen autosta tapahtuu vain tarvittaessa ja viiveellä.

Jos Pong jossain piileskelee, se piileskelee Tech 2:n muistikortilla. Tässä vaiheessa SaabLink pingasi vahvistuksen: tervetuloa lukemaan suljettua foorumia. Linkistä löytyi kaksi viestiä. Nimimerkki **spgeebee88** USA:sta kirjoitti vuonna 2008: ”Odottakaapas kunnes keksitte miten pelata Pongia Tech 2:lla.” **Lastaero** Australiasta vastasi: ”2004-, 9-3, Airbags, Miscellaneous… Meille näytettiin mistä se löytyy, kun läpäisimme Master Tech -kokeen muutama vuosi sitten.”

Niinpä rullasimme Skrollin testilaboratorioon vuosimallin 2007 Saab 9-3:n ja ryhdyimme skannaamaan. Kävimme läpi kaikki turvatyynyvalikot, moottorinhallintayksikön sekä lopulta kaikki muutkin valikot, eri ohjelmistoversioita ja keksimiämme kikkoja, mutta Pongia ei löytynyt. Onko tämä sittenkin maailmanluokan vedätys, jota kaikki pistävät eteenpäin?

Airbag-niminen valikko tosin löytyy vain vuosimallin 1999-2003 autoihin kytkeytyessä. Myöhemmin se on korvattu Airbags and Seat Belts -valikolla, josta vuosimallin 2007 Saab 9-3:een kytkeytyessä emme löytäneet Pongia. Kysyimme muutamilta tahoilta, mutta kukaan ei tiennyt. Myös Tech2 Owners’ club -Facebook-ryhmässä moni on etsinyt Pongia turhaan.

Ryhmässä kuitenkin **Mark Giourmetakis** muisteli, että se löytyy vain 2004-vuosimalliin kytkeytyessä: ”Pelasin sitä vuosia sitten 9-3 9440:lla (2004) Airbagsissa… se toimi vain sillä mallilla/vuodella.” Mark tosin jatkaa vieläkin villimmin: ”Kuulin, että yksi versio Pongista sisälsi alastoman tytön.”

Rikkinäinen puhelin

Suljetulta SaabLink-foorumilta löytyi lopulta nimimerkki **spgprojectin** kirjoitus, jonka perunakamerakuvat vahvistivat perustarinan. ”Toimii vain 9-3:ssa 2004:stä ylöspäin. Pääset siihen reittiä Body, Airbag-valikko, Miscellaneous ja sitten F2.” Kuvat eivät ole painokelpoisia, mutta niiden ja Tech 2:n tutkimisen perusteella replikoimme ”ruutukaappauksen” viereen.

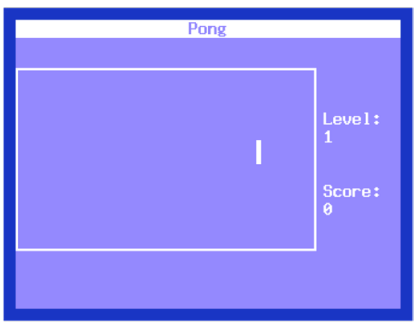
Spgprojectin viesti oli samalla kaikkein vanhin löydös, vuodelta 2007. Poissulkeaksemme silkan huijauksen, tarvitsisimme kuitenkin paremman lähteen. Onneksi Tech 2:n ohjelmisto tallennetaan sille muistikortille, eli sitä voi siis lukea. Kortti on lineaarinen, joten sisältö tulee ulos yhtenä pötkönä. Avasimme binääritiedoston heksaeditoriin ja annoimme hakusanaksi Pong…

”----- *Welcome to Pong ----- This is the Tech2 version of the classic game Pong. It is possible to move the ’racket’ with the arrow keys in all four directions. The closer to the left wall you are when you hit the ball the more points you will get. Have fun.....*”

Teksti löytyi suunnilleen samasta paikasta Tech 2 -ohjelmistoversioiden 135-148 binääristä, mutta ei vuosimallia 1999 vanhemmille autoille tarkoitetusta versiosta 44. Teksti näkyy myös spgprojectin kuvissa.

Pongia ei siis ollut piilotettu pingtoiminnon rinnalle, saati moottorinhallintayksikköön. Vuosimalliväitteet olivat myös epätarkkoja. Pong on ominaisuus tietyissä Tech 2 -skannerin ohjelmistoversioissa. Auton mallivuoteen se ei liity, mutta skannerin valikkorakenteesta johtuen pääsiäismuna vaikuttaa ilmestyvän vain sopivaan autoon kytkettäessä.

Vastaus otsikon kysymykseen on kyllä ja ei: Saabissa ei ollut Pongia, mutta Saabilla kyllä oli ”Pong” – ainakin piilossa binäärissä.



Saabin muistikortti

Saabin ja useiden muiden automerkkien käyttämän Motorola-pohjaisen Tech 2 -huoltoskannerin firmware-ohjelmisto ladataan vanhanaikaiselta PCMCIA-muistikortilta. Hyvänä puolena ohjelmisto on mahdollista ladata muistikortilta tarkempaan syyniin. Huonona puolena tällaisen kortin tulkkaaminen on tänä päivänä hankalaa.

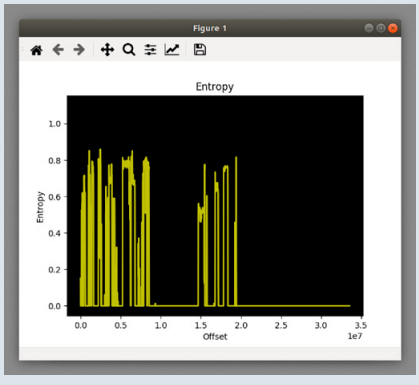
Lineaarinen PCMCIA-kortti

Yksistään PCMCIA-muistikorttipaikan tai -lukijan löytäminen on vaikeaa. Tämä digikameroiden aikaa edeltänyt formaatti kun ei lyönyt läpi tietokoneenkäyttäjien arkipäivään. Moderneista tietokoneista ei löydy enää mitään microSD-paikkaa suurempaa, jos sitäkään. Järeämmästä työraudasta löytyy ExpressCard-korttipaikkoja, mutta niihinkään PCMCIA-kortit eivät suoraan sovi.

Lisäksi Tech 2 käyttää niin kutsuttua lineaarista muistikorttia. Toinen yleinen PCMCIA-muistikorttien tyyppi ovat ATA-kortit, jotka muistuttavat tietokoneen kannalta kiintolevyä. Saabin muistikorttia lukeakseen tarvitaan lineaarisia PCMCIA-kortteja lukeva korttipaikka, ja ne ovat vieläkin harvinaisempia. Jännitteenkin kanssa saa olla tarkkana – lukijan (ja kortin) pitäisi tukea 5V/12V-tyyppiä.

Yleinen suositus lukijaksi on vanha kannettava Texas Instrumentsin piirisarjalla. Tosin kortin kytkeminen koneeseen ei vielä tee mitään – helposti avattavaa tiedostojärjestelmää kun ei ole. Tässä auttaa *Elan Memory Card Explorer* (synchrotech.com). Uudemmille koneille löytyi ExpressCard-sovitin *SCM ADE ExpressCard 34 Adapter für PCMCIA/3,3V 5V 12V/SRAM/Linear Flash/ATA* ja *Cardware*-ohjelmisto (scm-pc-card.de) sekä *CSM OmniDrive* USB:hen (csm.de).

Huomionarvoinen on myös *AleXRE Tech2 Card Writer* -ohjelma, jolla voi kirjoittaa Tech 2:ssa olevalle muistikortille. Näillä ohjelmisto saadaan ulos ja sisään yhtenä 10–32 megatavun binääritiedostona.



Binwalk: Saabin muistikortin entropia.

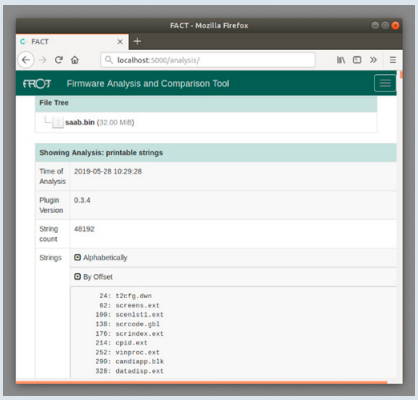
Motorola MC68332

Seuraava haaste on seurausta Tech 2:n suoritinraudasta, joka sekin edustaa nykymittapuulla harvinaisempaa teknologiaa. Laitteen sydämenä sykkii Amigan, Atari ST:n ja varhaisten Macien suorittimen sulautettu sukulainen: Motorola (nyk. Freescale) MC68332-suoritin tai oikeastaan mikrokontrolleri, jossa on käskykannaltaan hieman muovattu CPU32-ydin.

Tämä tarkoittaa sitä, että suurin osa reversointityökaluista tipahtaa harkinnasta ainakin osittain: *Radare2*, *BinaryNinja*, *Hopper*… MC680x0-perheelle saattaa vielä löytyä tukea ja ohjelmissa voi olla muita hyödyllisiä ominaisuuksia, mutta MC6833x-disassembleria ei.

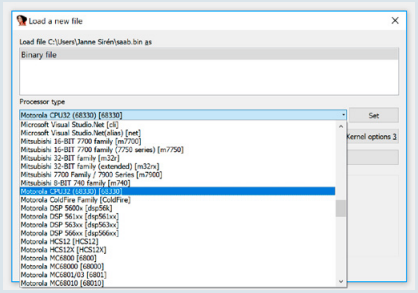
Binwalk-firmware-analysaattorilla (github.com/ReFirmLabs/binwalk) on kuitenkin hyvä aloittaa. Sillä voi arvioida tiedoston rakennetta monesta eri kulmasta, kunhan asentaa eri laajennukset, kuten *python-lzma*, *python-crypto* ja *pyqtgraph*. Windows-käyttäjien kannattaa suosiolla asentaa rinnalle Linux (esimerkiksi Ubuntu) vaikkapa Oraclen maksuttomaan VirtualBoxiin (virtualbox.org), sillä binwalkin laajennusten asennusoperaatio on huomattavasti helpompi Linuxin pakettivarastojen kanssa.

Binwalkilla voidaan tunnistaa, purkaa ja erotella ohjelmistosta erilaisia osia. Sillä voidaan myös selvitellä, onko firmware-binääri esimerkiksi pakattu tai kryptattu sekä etsiä ohjelmistokoodin alkua disassembleria varten. Koska binwalk voi olla vähän pelottava, *Firmware Analysis and Comparison Tool* (github.com/fkie-cad/FACT_core, entinen Fraunhofer's) tarjoaa käyttäjäystävällisemmän käyttöliittymän.



FACT: Saabin muistikortin merkkijonot.

Esivaihe on tärkeä sen viimeisen askeleen kannalta: reversoinnin. Mikäli mielii tutkivansa tai muuttavansa Saabin ohjelmiston koodia – vaikka sitä kuuluisaa Saabin Pongia – belgialaisen Hex-Raysin (hex-rays.com) kaupallinen *IDA* (*Interactive Disassembler*) *Starter* ja *Pro* ovat parhaita, joskin hintavia työkaluja. Ne tukevat myös MC68330-perhettä (CPU32).



IDA Pro: MC6833x disassembler.

Loppusanat

Tänä keväänä Mercedes-Benz esitteli uuden automallinsa viihdejärjestelmää ratilla pelattavalla *SuperTuxKart*-pelillä. Peliominaisuus saattaa päästä jossain muodossa jopa tuotantoon asti. Tesla taas on lupaillut Atari-peliensä seuraksi Unreal/Unity-tuotoksia.

Saab oli piilo-Ponginsa kanssa edelläkävijä. Mitä muuta voikaan odottaa pieneltä kapinalliselta, joka kehitti 1990-luvulla muun muassa joystickilla ajettavan auton ja liimasi tuotteisiinsa tarran: "Made in Trollhättan by Trolls."

Talentec Bartop Kit ja RetroPie

Pelikabinetti kahvihuoneeseen

Jari ja Atte kertovat Maria 01 -yhteistoimistolle hankitusta pelikabinetista – sopii niin konttorille kuin kotiinkin.

Teksti ja kuvat: Jari Jaanto, Atte Penttinen, Janne Sirén

Maria 01 -startup-keskittymässä Helsingissä on töissä paljon nuoria aikuisia tai aikuisia, jotka viettivät nuoruuden pelaamalla pelihalleissa. Niinpä ajattelimme toimistolla, että paras tapa rentoutua työpäivän keskellä olisi ottaa nopea tauko pelihallien inspiroimalla pelikabinetilla.

Yleensä nuorilla uusyrityksillä ei ole mahdollisuutta panostaa kalliisiin hyvinvointipalveluihin, joten tämä olisi yksi kohtuuhintainen väline työhyvinvoinnin parantamiseen.

Ikeaa ilman ohjeita

Pienen tutkimustyön jälkeen löysimme Espanjasta sopivan pelikabinettitoimittajan Talentecin, jolta löytyi projektiin sopiva tuote. Tilasimme Talentecilta kabinetin räätälöidyillä mitoilla. Halusimme tuoda siihen myös ripauksen historian havinaa, joten tilasimme sen Pac-Man-teippauksilla. Talentecilla teippaukset voi valita valmiista vaihtoehdoista tai itse toimitetuista.

Laitteisto rakennettiin käyttäen Raspberry Pi -pientietokonetta. Ohjaimiksi valittiin pallomalliset ohjaimet, koska nämä muistuttivat eniten lapsuudessa koettuja kapuloita. Ohjelmistoksi taas valikoitui RetroPie, joka tukee useampaa emulaattoria. RetroPien saa ladattua helposti verkosta ja sen voi asentaa tietokoneelta microSD-muistikortille, jota Raspberry Pi käyttää sitten massamuistina.

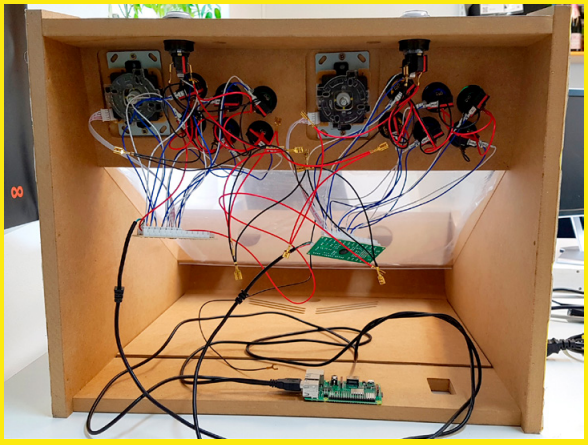
Lähetystä odotettiinkin kuin pikkulapsi joulua. Parin viikon jälkeen paketti saapui, ja onneksemme Ikea-kalusteiden kokoaminen on tällaisessa toimistohotellissa tuttua. Ohjeistus sai tosin Ikean ohjemonisteet näyttämään loistavilta: Bartop Kitin mukana oli kaksipuoleinen A5-ohje, jonka toisella puolella oli värillinen kuva siitä, miltä kabinetin pitäisi näyttää ja toisella puolella kuvat osista…

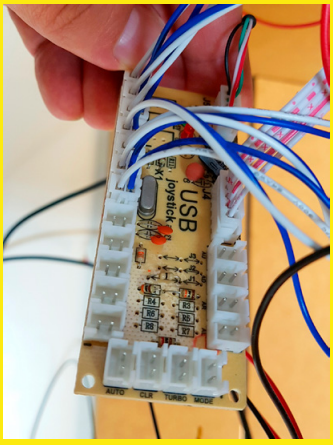
Ei muuta kuin ruuvimeisseli käteen ja yrittämään. Muutaman yrityksen ja erehdyksen jälkeen kabinetti saatiin kuin saatiinkin kasaan. Tässä vaiheessa olisi ollut hyvä miettiä hieman softaa kabinetissa, myöskin teippaus olisi ollut huomattavasti helpompaa ennenkuin osat oli naulattu yhteen. Mutta ainakin oppimiskäyrä oli nopea. Eikä kabinetin kasaaminen sinänsä vaikeaa ollut. Kokoaminen vei yhteensä pari tuntia ja teippaaminen toisen mokoman.

Elektroniikan kasaaminen meni sujuvasti. Mukana tulleet Cy-822A-malliset kytkinlevyt toimivat peliohjainten ja Raspberry Pi'n välissä. Cy-822A-levyistä lähtee suora USB-kytkentä Raspberry Pi'hin ja RetroPie tunnisti ohjaimet automaattisesti. Hommaa helpotti myös se, että RetroPin mukana tuleva EmulationStation-käyttöliittymä osaa kytkeä ohjainten nappulat halutun mukaisesti. Näin saimme esimerkiksi "Pelaajan valinta" -napit vastaamaan kolikon sisään syöttämistä.

Monitoriksi valikoitui varastosta napattu vähällä käytöllä ollut 19 tuuman LCD-näyttö, ja ääniä varten pelikabinetin sisään upposivat minikaiuttimet. Pelikabinetti onkin palvellut hyvin hektisen yhteistoimiston arjessa.

Talentec löytyy internet-osoitteesta talentec.es *ja RetroPie osoitteesta* retropie.org.uk.





Historiallisen Marian sairaalan tiloissa Helsingissä toimiva Maria 01 (maria01.co) kertoo olevansa Suomen suurin keskittymä startup-yrityksille. Entinen sairaala toimii työpaikkana reilulle tuhannelle teknologia-alan henkilölle ja sen tiloissa järjestetään vuosittain 300 teknologiatapahtumaa.

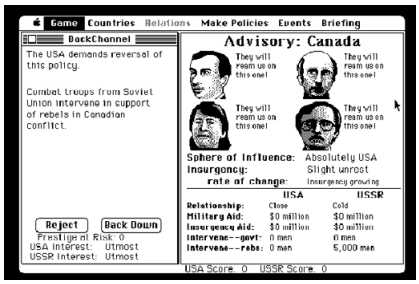
JOKIN PELIPALSTA

VALLOITUSPELIT

Olisinpa maailmanvaltias

Matka maailmanvaltiuteen tuppaa olemaan pitkä, kivinen, työläs ja useimmille mahdoton. Onneksi sitä voi edes vähän simuloida pelimuodossa.

Teksti: Jukka O. Kauppinen Kuvat: Jukka O. Kauppinen, Mobygames

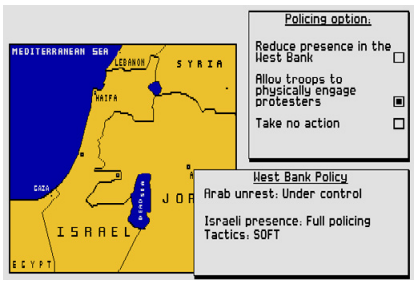


Balance of Power

Mindscape, 1985
Amiga, Apple II, Atari ST, PC, Mac, MSX, PC88, PC-98

Legendaarisen pelisuunnittelija **Chris Crawfordin** *BoP* on monella tapaa kuin edelleen viilattu *Geopolitique*. Myös se astuu kylmän sodan ytimeen suurvaltapoliittisen strategian keinoin. Hassua sinänsä, että näiden kahden pelin välillä on näinkin lyhyt aika, sillä *BoPin* graafinen toteutus ja hiirikäyttöliittymä tekevät siitä huimasti pelattavamman ja ymmärrettävämmän.

BoP on nykystandardeilla edelleen täysin pelattava, mutta myös työläs ja turhauttavan vaikea. Vallan ja mahdin houkutus on silti lumoavaa. *Bop* ja *Geopolitique* olivat aikalaisikseen harvinaisia esimerkkejä avoimista pelimaailmoista, joiden tapahtumat kulkivat joka kerralla eri suuntiin ilman ennalta määrättyjä kaavoja.



Conflict: The Middle East Political Simulator

Virgin Mastertronic, 1990
Amiga, Atari ST, PC

Israel on aina hyvä, kiinnostava ja kuuma aihe. Jokseenkin vihamielisessä ympäristössä sijaitseva miniatyyrisuurvalta on kamppaillut olemassaolostaan itsenäisyysjulistuksestaan lähtien sekä sotilaallisella että diplomaattisella taistelukentällä. Tämä poliittis-militaristinen strategiapeli tiivistää hyvin oleellisen: miten tasapainoilet naapurien kanssa? Miten paljon panostat sotilaalliseen voimaan, kulissientakaiseen kieroiluun tai ydinaseohjelmaan? Mitä teet, jos tilanne rajalla kuumenee tai räjähtää? Ja sitten vielä ne palestiinalaiset…

Conflict paketoi Israelin dilemmat kiinnostavaksi peliksi, jota on helppo pelata vaikkapa Archive.orgin selainemulaattorin kautta. Siitä on myös vähän modernimpi freeware-versio, jonka voi ladata Windowsille täältä: www.gamefront.com/files/conflict-m-e-p-s/download.

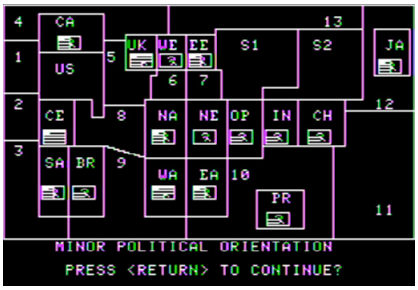
Twilight Struggle

Playdek / Asmodee, 2016
Android, iOS, Mac, PC

Samannimiseen lautapeliin pohjautuva videopeli on maukas ja kiinnostava moderni tulkinta maailmanvalloituksesta sanoin, kirjein ja panssariarmeijoin. Kortti- ja lautapelin aineksia sekoittava strategia kuvaa skenaariosta riippuen kylmän sodan alkamisen ja nykyajan välistä ajanjaksoa.

Modernimman toteutuksensa ansiosta peli on oikein pelattava ja selkeä. Se kuvaa maailmaa varsin pikkutarkasti mutta myös selkeästi niin, etteivät pelaajat huku infoähkyyn. Muihin maihin vaikuttaminen ja niiden imaisu omaan leiriin ja ideologiaan on lähes julman koukuttavaa, sillä mikäs sen hienompaa kuin Suomen ”vaikuttaminen” punaleirin täysimittaiseksi jäseneksi. SKP, 100 prosenttia äänistä.

Kannattaa muuten pistää Steamissä wishlistille, sillä tämän saa napattua parhaimmillaan muutamalla eurolla omaan pelikirjastoon.



Geopolitique 1990

SSI, 1983
Apple II, Commodore 64

SSI:n kylmän sodan simulaatio oli yksi ensimmäisistä, ellei jopa aivan ensimmäinen massiivisen mittaluokan supervaltasimulaatio, jossa pelaaja astui Yhdysvaltain ylivaltiaaksi tavoitteenaan nujertaa Neuvostoliitto kauppa ja tuotanto, politiikan tai sotilaallisen mahdin mittareilla.

Suunnittelija ja ohjelmoija **Bruce A. Ketchledge** ehti ilmeisesti luoda aikoinaan vain kaksi strategiapeliä, mutta molemmat olivat omaperäisiä ja kiinnostavia luomuksia. Tosin tänään ne ovat myös melkoisia dinosauruksia, sillä täysin tekstipohjaisten strategioiden pelaaminen vaatii ohjekirjojen lukemista ja pitkää pinnaa. Silti täytyy arvostaa sitä, että jo näin varhaisessa kotimikroiluvaiheessa kuusneloselle ja Applelle tehtiin jotain näin monimutkaista, syvällistä ja vaikuttavaa. Grand strategy -sotasimulaatio näet mallinsi todellista maailmaa sen pelottavimmilla tasoilla ja mahdollisuuksilla, eikä suuremmin rajoittanut pelaajaa tämän pyrkimyksissä. Ja jos lopulta oikein huonosti kävi, niin sitten peli siirtyi politiikasta taistelukentälle. Kuka tässä maailmassa lopulta määrää, häh?

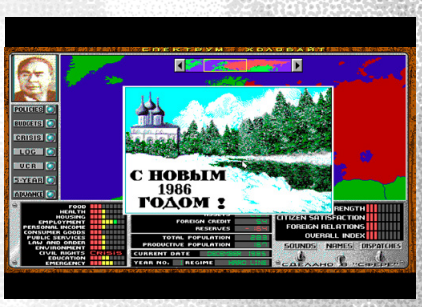
Global Conquest

Ozark / Microplay, 1992
PC

Dan(i) Bunten, yksi historian tärkeimmistä pelisuunnittelijoista, loi monia mullistavia moninpelejä (kuten MULEn, ks. Skrolli 2017.1), joista *Global Conquest* lienee sieltä vähemmän muistetusta päästä. Tämän artikkelin mitassa se on myös outolintu, sillä pelissä itsessään on vain niukalti diplomaattisia toimintoja. Silti diplomatiaa riittää, sillä *Global* suunniteltiin nimenomaan 2–4 hengen moninpeliksi, jossa suorat linjat käyvät kuumina pelaajien välillä. Tekoälypelaajiakin toki voi käyttää vastustajina.

Buntenin edelläkävijän aivot kuplivat ja yrittivät tehdä pelistä myös tasapainoisen, joten esimerkiksi satunnaiset tapahtumat tuppaavat auttamaan häviöllä olevia komentajia ja varovasti kampittamaan pistetilastojen johtajia. Häviölläkin olevalla pelaajalla on siis aina edes pieni mahdollisuus. Strategiassa oli myös useita erilaisia skenaarioita ja maailma luotiin nollasta uniikiksi joka pelikerralle.

Vaikka sotailu olikin suhteellisen suoraviivaista ja nopeatempoista, niin monet hyvät ja uudet ideat tekivät *Globalista* yhden aikansa erikoisemmista julkaisuista. Moninpelaaminen modeemilla ja lähiverkossa ei ollut vielä kovinkaan yleistä, mutta Bunten puski sitkeästi yhteisöllisen pelaamisen sanomaansa eteenpäin, mielestäni varsin onnistuneesti. Nykyisillä DOSBox-virityksillä (ja pelin 2.0-päivityksellä) globalismi toimii myös internetin yli, mutta kyllä tätä kannattaa myös retrolaneilla kokeilla.



Crisis in the Kremlin

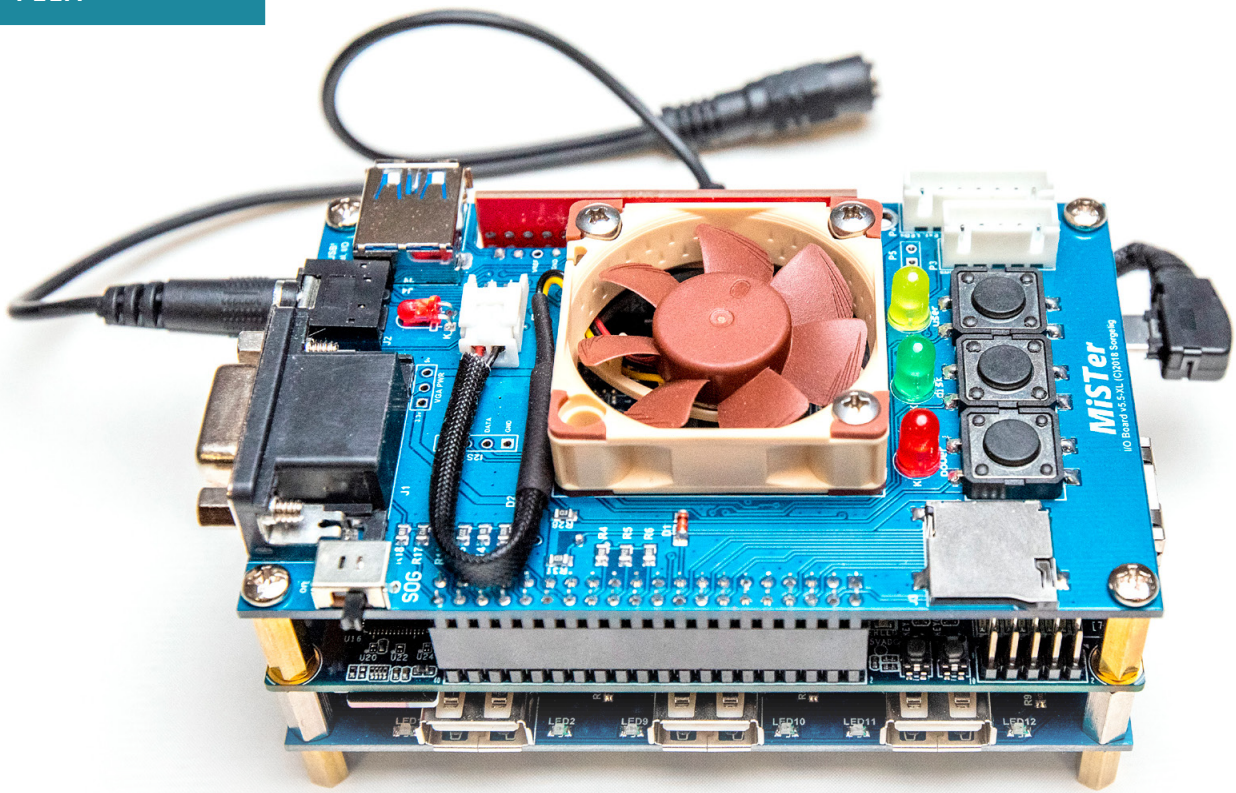
Spectrum Holobyte, 1992
PC

Tässäpä todella omituinen retrohelmi. Neuvostoliiton hajoaminen vuonna 1991 räjäytti suurvaltapoliittisen kentän täysin uusiksi, mikä taisi yllättää myös *Crisisin* tekijät. Todellisuus kun rynnäköi pelisuunnittelijoiden edelle, minkä myötä Neuvostoliiton kriisiaikoja kuvaavan poliittisen simulaation pelaaminen olikin varsinaista jälkijossittelua. Olisiko Neuvostoliitto voinut sittenkin säilyä – ja jos, niin millä hinnalla?

Crisis hyödyntää taitavasti todellista historiaa ja poliittisia raameja, joissa eri linjauksia edustavat puolueet johtajineen yrittävät luovia. Mennäänkö kovalla kommunismilla vai avataanko rajat ja toivotaan parasta? Miten vastustajat suhtautuvat päätöksiin? Saadaanko niitä edes nuijittua läpi? Tulevatko panssarit työhuoneen ovelle toteamaan, että ei näin toveri **Kauppishka**, ei näin?

Vuodesta 1985 alkava presidenttisimulaatio on haastava ja jopa realistisen turhauttava peli. Kansalla on nälkä, jano ja aina jokin kriisi räjähtää naamalle. Pienet tasavallat pyristelevät irti punakarhun syleilystä ja yskivän talouden kohentaminen vaatii radikaaleja liikkeitä. Mutta ehkä huominen on parempi, kun lähetetään panssarit Tallinnan kaduille rauhoittelemaan itsenäisyydestä mekastavia mielenosoittajia.

Crisis sopii hienosti myös maailmanparantajakurssien työvälineeksi, sillä sen kautta oppii erittäin konkreettisesti maailmanpolitiikan syy-seuraussuhteita ja erilaisten voimaryhmittymien välistä tasapainottelua. Ylipäätänsäkin ajatus rapautuvan suurvallan johtamisesta ja paremman huomisen luomisesta kuulostaa yllättävän tutulta myös nykypäivän politiikkaa peilatessamme. Kenties *Crisis in the Kremlin 2020 Edition* olisi pian paikallaan?



MiSTer – kasa klassikoita kerralla

Uudelleenohjelmoitavat FPGA-piirit ovat monin tavoin mullistaneet retrokoneiden harrastamisen. MiSTer-projekti pyrkii luomaan edullisen, avoimen ja laajennettavan kokonaisratkaisun monenlaisten laitteiden matkimiseen.

Teksti: Mikko Heinonen Kuvat: Janina Joutsen, Mikko Heinonen

Jos on tullut viime aikoina seurailleeksi etenkin vanhemmalle tietotekniikalle valmistettuja lisälaitteita, lienee ollut vaikea välttyä kirjainyhdistelmältä FPGA eli *Field Programmable Gate Array*. Mikropiiri, jonka sisään voidaan toteuttaa jopa useita toisia mikropiirejä on konseptina juuri niin kätevä kuin miltä se kuulostaa, ja se onkin yksinkertaistanut monenlaisten laitteiden suunnittelua.

Prototyyppien luonti ja laitteiden päivitys helpottuu merkittävästi, kun FPGA voidaan ohjelmoida uudelleen vaikka jokaisella käynnistyskerralla käyttäen erityistä VHDL-kuvauskieltä (*VHSIC Hardware Description Language, VHSIC = Very High Speed Integrated Circuit*). Varsinaista tuotantoa varten FPGA:n pohjalta voidaan sitten luoda ohjelmoinniltaan pysyvä ASIC-piiri (*Application Specific Integrated Circuit)*, jonka sarjavalmistus on halvempaa. Yhä suositumpaa on myös toimittaa lopullinenkin laite FPGA:lla varustettuna.

Retrolaitteiden harrastajat ottivat FPGA:t ja niitä hieman yksinkertaisemmat CPLD:t (*Complex Programmable Logic Device*) ilolla vastaan niiden saapuessa markkinoille parikymmentä vuotta sitten. Piensarjoina toteutettavia laitteita olivat varsin usein piinanneet suunnitteluvirheet, joiden korjaaminen jälkikäteen oli joko hankalaa tai täysin mahdotonta. Ohjelmoitava piiri on tässä suhteessa huomattavasti anteeksiantavampi, sillä ”kytkentöjä” voidaan muuttaa jälkikäteen ja näin korjata ongelmia sekä lisätä ominaisuuksia. Samoin piirilevysuunnittelu muuttuu yksinkertaisemmaksi, sillä CPLD/FPGA pystyy korvaamaan lähes kaikki aktiivikomponentit.

Suosittuja CPLD/FPGA-pohjaisia lisälaitteita ovat muun muassa Commodore 64:n levyaseman korvaava 1541 Ultimate jälkeläisineen (mm. Skrolli 2015.1) ja Nintendo Entertainment Systemiin RGB-videolähdön lisäävä NESRGB (Skrolli 2016.1). Samoin konsoliharrastajien suosimat Everdrive-kortinlukijat (Skrolli 2015.3) käyttävät uudelleen ohjelmoitavia piirejä. Usein onkin niin, että pientä oheistoimintoa toteuttava piiri voisi helposti matkia koko palvelemaansa laitetta. Vielä oma lukunsa ovat Vampiren (Skrolli 2014.1, 2018.2) tyyppiset FPGA:lla toteutetut Amiga-kiihdytinkortit.

Kulti, kutistin koneesi

Oheislaitteiden synnyttämisen lisäksi on luonnollisesti mallinnettu myös kokonaisia tietokoneita ja konsoleita. **Gideon Zweijtzer**, joka 1541 Ultimaten jälkeen loi Ultimate-64:n eli FPGA-kuusnelosen (ks. Skrolli 2018.3), totesikin tämän olleen hänen tavoitteensa alusta alkaen. Ostava yleisö oli kuitenkin alkujaan kiinnostuneempi levyaseman korvikkeesta. Kenties ajan kuluminen oli hankkeelle hyväksi, sillä rauhassa kypsytelty Ultimate-64 on yksi harvoja entusiasteille kelpaavia FPGA-mikroja ja miltei täysin yhteensopiva Commodore 64:n kanssa. Saavutuksen arvoa nostaa erityisesti se, että versioiden runsauden vuoksi läheskään kaikki aidotkaan Commodore 64:t eivät ole täysin Commodore 64 -yhteensopivia.

Ultimate-64 ei kuitenkaan ollut ensimmäinen kaupallinen FPGA-kuusnepa, sillä **Jeri Ellsworth** ja **Jens Schönfeld** toivat jo vuonna 2003 markkinoille C-One-nimisen uudelleen konfiguroitavan kasibittitietokoneen, joka oli ajateltu C64:n päivitykseksi. Se ei ollut puhdas FPGA-toteutus, vaan käytti ohjelmoitavien piirien rinnalla 65816-suoritinta, siis kuusnelosta tah-

dittavan MOS 6510:n 16-bittistä versiota.

Koska retroraudan hinta 2000-luvun alussa oli hyvin toisella tasolla kuin nykyään, markkinat eivät ehkä olleet vielä valmiit usean sadan euron hintaiselle C-Onelle. Implementaation tarkkuuskaan ei ollut asiaan perehtyneitä tyydyttävällä tasolla, joten varsinainen läpimurto jäi tekemättä. C-Onelle ilmestyi myöhemmin muun muassa Amstrad CPC-, Spectrum- ja Amiga-coret sekä rautalaajennuksia, joilla käyttöön saatiin lisää FPGA-tilaa. Jerin tähän työhön pohjautunut erillishanke, halpa ASIC-pohjainen C64 DTV -minikonsoli (Skrolli 2018.2) sentään myi varsin hyvin – Amerikan Ostos-TV, QVC myi niitä kuusinumeroisen määrän.

Commodoren uudempi kotimikro Amiga on teknisesti jonkin verran monimutkaisempi laite, joka sisältää erilaisia erikoispiirejä suorittimen avuksi. Ensimmäinen Minimig-nimellä tunnettu FPGA-Amiga olikin tavallaan puolittainen toteutus, sillä se käytti aitoa Motorolan 68000-suoritinta ja vain erikoispiirit oli toteutettu Xilinxin FPGA:lla. Tämä tarjosi kohtalaisen hyvän yhteensopivuuden jo tuolloin saatavilla olleilla piireillä, mutta toisaalta se myös rajoitti laitteen nopeuden laajentamattoman Amiga 500:n tasolle.

Ensimmäinen Minimig-laite soveltui parhaiten lähinnä vanhojen pelien pelaamiseen ja jäi sekin korkean hintansa vuoksi marginaalituotteeksi. Avoimena lähdekoodina toteutettu Minimig-core on kuitenkin sittemmin kehittynyt pitkälle. **Dennis van Weerenin** käynnistämä projekti on käännetty tehokkaammille FPGA-piireille, ja yhdistettynä 68000-suoritinperheen FPGA-kuvaukseen Minimig mahdollistaa nykyisin kokonaisen Amigan mallinnoksen ilman erillistä suoritinta. Eräs ensimmäisiä laitteita, jolle se sovitettiin, oli juuri C-One. Jens Schönfeldin uudempi FPGA-hanke taas on nimeltään Turbo Chameleon (Skrolli 2015.1, 2018.2), joka toimii sekä itsenäisenä C64:nä että megaturbona, jonka voi liittää oikean kuusnepan käyttäjänporttiin. Myös muita kasibittikoneiden coreja sekä kevennetty Minimig on käännetty Turbo Chameleonille.

Commodore-harrastekenttä ei ollut luonnollisesti yksin pyrkimyksineen, vaan myös muiden kotimikrojen parissa tapahtui. Vuonna 2006 julkaistu One Chip MSX kuvasi MSX2-mikron Altera Cyclone -piirillä. ZX-UNO puolestaan on eräs monista Sinclair Spectrum -toteutuksista.

Pelaajatkin havahtuvat

Myös pelikonsolipuolella on viime vuosina herätty FPGA:n ilosanomaan. Tärkeä motivaattori työlle on ollut saada vanhat kuvaputkiaikaan suunnitellut pelit toistumaan somasti myös teräväpiirtona, ilman huippusuoritusta häiritsevää viivettä. NES ja SNES Minin kaltaiset kaupalliset toteutukset ovat enimmäkseen halpoihin järjestelmäpiireihin perustuvia, Linux-pohjaisia emulaattorikonsoleita, mutta perfektionistit ovat etsineet täydellistä kokemusta hyödyntämällä FPGA:n mahdollisuuksia. Oikein toteutettuna FPGA mahdollistaa laitteen matkimisen huomattavasti ohjelmistopohjaista emulaatiota tarkemmin.

Tunnetuin FPGA-konsoleiden valmistaja on Analogue, jonka Super Nt- ja Mega Sg -nimiset uusintaversiot Super Nintendosta ja Sega Mega Drivestä ovat saaneet yläpeukkua myös aiheeseen vakavammin suhtautuvilta harrastajilta. Kääntöpuolena on Analogue-konsoleiden korkeahko hinta: pelkästä konsolista saa maksaa Suomeen toimitettuna yli 200 euroa, eikä tämä edes sisällä ohjaimia.

Kaikki maailman coret, yhtykää!

Vaikka erilaisten FPGA-laitteiden kenttä on siis kohtuullisen hajanainen, kyse on periaatteessa samasta asiasta: piirien mallintamisesta toisilla piireillä. Ei ole mitään teoreettista estettä sille, etteikö tietylle FPGA:lle kirjoitettu VHDL-koodi olisi käännettävissä toisellekin mallille, vaikka eri valmistajien tuotteiden välillä on luonnollisesti eroja. Kyse onkin enemmän siitä, että pyrkiessään saamaan hinnan mielekkääksi erilaisten harrastajalaitteiden valmistajat ovat yleensä valinneet käyttöön piirin, jonka tila riittää juuri heidän tarvitsemaansa käyttöön. Mitä tehokkaampi ja monipuolisempi eli ”suurempi” FPGA valitaan, sitä enemmän se myös maksaa.

Monenlaisessa retrosopassa keitetty puolalainen **Przemyslaw Krawczyk**, myös **Lotharekina** tunnettu, saikin idean tällaisesta yleislaitteesta jo jokunen vuosi sitten. Projektin lopputuloksena syntyi MiST, alkujaan ”Mini ST”, joka tarjoaa kirjoitushetkellä jonkin tasoisen tuen 25 tietokoneelle, 11 konsolille ja seitsemälle kolikkopelille. Koneeseen ladataan käynnistyksen yhteydessä haluttu core SD-kortilta, ja se tukee USB-näppäimistöä, VGA-näyttöä ja perinteisiä 9-nastaisia joystick-liitäntöjä.

Reilun 200 euron hintainen MiST on jo kypsässä versiossa 1.4 ja etenkin sen Atari ST -tuki on hyvin toimiva. Amiga-tuki perustuu – ei niin yllättäen – Minimig-coreen. MiST oli valinta esimerkiksi Suomen pelimuseon Amiga-pelejä pyörittämään, ja se onkin toiminut tässä tehtävässä luotettavasti.

Pari kirjainta, iso ero

Eräs MiST-projektin aktiivinen osallistuja on nimimerkin **Sorgelig** taakse kätkeytyvä, ilmeisesti venäläissyntyi-

Kun MiSTeriin kytkee virran, minimaalinen GNU/Linux käynnistyy ja kysyy käyttäjältä, mitä matkittaisiin tänään. Taustalla on tyylikäs tv-lumisade, joka tosin sekoittaa videokaapparit.

nen kehittäjä, joka on tehnyt laitteelle puoli tusinaa corea. Hänen erityistä osaamisaluettaan ovat ZX Spectrum variantteineen (mukaan luettuna erikoinen SAM Coupé, ks. Skrolli 2015.3) ja itäblokin mikrotietokoneet.

Juuri Sorgelig sai alkukesästä 2017 idean kääntää MiSTille luodut coret isommalle, mutta samalla edullisemmalle FPGA:lle. Syntyi MiSTer-projekti, joka hyödyntää Terasic DE10-Nano -piirilevyä. Se on reilun sadan euron hintainen kortti, jota Intel käyttää omassa FPGA University -koulutusohjelmassaan. Hinta on saatu painettua alas laitteistosponsorien (muun muassa Analog Devices, Würth Elektronik ja Panasonic) ansiosta, mutta kortilla oleva rauta on täyttä tavaraa: sen FPGA, Altera Cyclone V SE, on kooltaan yli nelinkertainen MiSTissä käytettyyn nähden (110 000 vs. 25 000 logiikkaelementtiä). Tukena samalla piirillä on 800 megahertsin ARM-suoritin, joka FPGA:n ohjaamisen lisäksi voi käsitellä sen puolesta myös erilaisia tehtäviä ja jopa emuloida ulkopuolisia lisälaitteita. Laitteessa on HDMI-lähtö ja se skaalaa kuvan aina teräväpiirtotarkkuuteen. Verkkokortti mahdollistaa tiedostojen siirron ja etäyhteyden koneeseen. Piirilevy sisältää niin paljon komponentteja niin pienessä paketissa, ettei sen valmistaminen harrastajavoimin todennäköisesti edes onnistuisi.

Tehon ohella vähintään yhtä kiinnostavaa on DE10-Nanon hyvä laajennettavuus: kortti on täynnä erilaisia pinnejä, joihin on mahdollista rakentaa omia rautalaajennuksia. Yksi oleellisimpia niistä on SDRAM-laajennus, joka mahdollistaa uudempien kotitietokoneiden RAM-muistin matkimisen. Kortilla on valmiina gigatavu DDR3-muistia, mutta koska myös ARM käyttää sitä yhtä aikaa FPGA:n kanssa, latenssi on liian suuri EDO RAMin simulointiin. Ilman SDRAMiakin on kuitenkin mahdollista pelata vaikkapa vanhoja kolikkopelejä, joiden muistitarve on joitakin kymmeniä kilotavuja.

Samalla tavoin merkittävä on I/O-kortti, joka asentuu MiSTerin päälle. Se tarjoaa RGB-liitännän, joka tuottaa täsmälleen alkuperäisen kaltaista kuvasignaalia, ulkoisen äänilähdön, helpomman pääsyn emolevyllä oleviin ohjauspainikkeisiin, toisen SD-korttipaikan sekä tärkeänä lisänä tuulettimen. Etenkin Minimigin kaltaiset monimutkaiset coret ajavat DE10-Nanon FPGA:ta sen verran lujaa, että lisäjäähdytys on tarpeen.MiSTer tukee USB-lisälaitteita, mutta DE10-Nanolla on vain yksi microUSB-liitin. Siihen voi joko hankkia kaupasta aktiivisen hubin tai sitten rakentaa USB Hub -kortin, jolla portteja saa käyttöön kerralla jopa kahdeksan kappaletta.

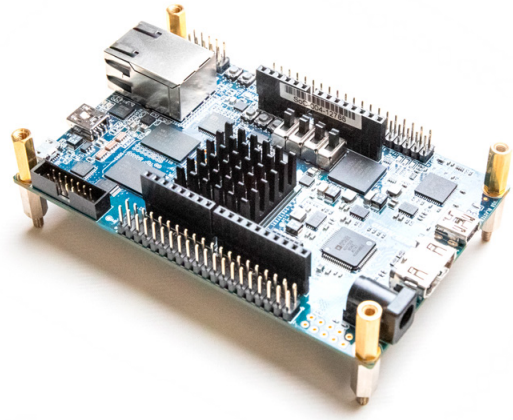
Tee se itse – tai älä

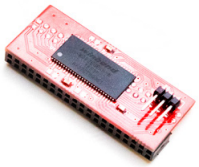
Kaikkien lisäkorttien piirustukset ja piirilevyjen tekoon tarvittavat Gerber-tiedostot löytyvät Githubissa sijaitsevasta MiSTer-wikistä. Onkin täysin mahdollista hankkia pelkkä DE10-Nano valmiina ja koota halutut lisäkortit itse. Hakemalla ”mister” suositun PCBWay-sivuston valmiista projekteista löytää yleensä uusimmat eri korttien versiot ostoskoriin lisättäväksi.

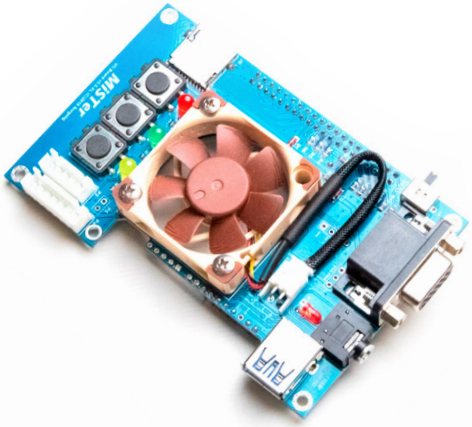
Osien keräily vaatii hieman enemmän vaivannäköä. Komponenttiluettelot ovat kohtuullisen helposti saatavilla, mutta useimmat elektroniikkatukkurit eivät myy yksittäisiä pintaliitoskomponentteja – esimerkiksi vastusten pienin tilausmäärä saattaa olla 100 kappaletta, vaikka yhden kappaleen hinta ei toki päätä huimaa. Käytännössä onkin järkevää koota yhteen porukka ja teettää sekä koota kerralla useampi piirilevy.

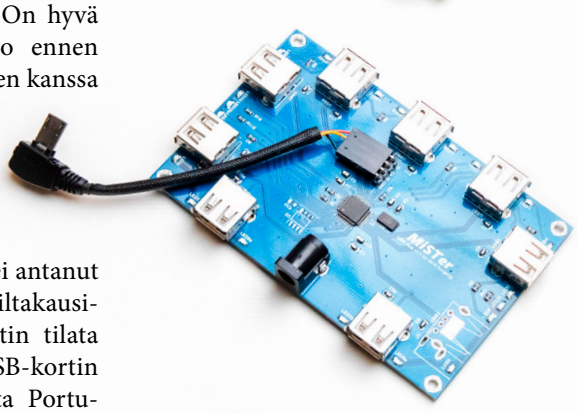
Lisäkorttien juotokset eivät ole yhtä pieniä kuin DE10-Nanolla, mutta vaativat joka tapauksessa edistyneen harrastajan kolvin varteen. On hyvä osata pintaliitostekniikka jo ennen kuin sitä alkaa näiden korttien kanssa harjoitella.

Kuten arvata saattaa, valmiiksi koottujen korttien ympärille on jo syntynyt pienimuotoista liiketoimintaa. Koska lehden deadline ei antanut periksi osien odottelulle ja iltakausien kolvausoperaatiolle, päätin tilata oman SDRAM-, I/O- ja USB-kortin RetroShop.pt-verkkokaupasta Portugalista. Hinnaksi tuli yhteensä reilut sata euroa, itse tehtynä selviää varmasti noin kolmanneksella tästä – siis jos ei laske omille työtunneilleen hintaa. Itse DE10-Nanon yksityisasiakas voi ostaa helpoimmin Amazon-verkkokaupasta: elektroniikkatukkureilla hinta on halvempi, mutta monet niistä vaativat Y-tunnuksen ennen kaupankäyntiä.









Ylhäältä lukien: MiSTerin aivot eli Terasic DE10-Nano, SDRAM-moduuli, I/O-kortti ja USB-kortti. Cyclone V SE -piirin päällä on jäähdytyssiili.

Kynnys retroiluun alenee mukavasti, kun sama pieni kortti voi olla vaikkapa Super Nintendo (ylävas.), PC Engine (yläoik.), Macintosh Plus (alavas.) ja SAM Coupé (alaoik.)

Lataa ja varmista

Kun käsissäni oli toimintavalmis MiSTer, oli aika kokeilla jonkin lataamista sen sisään. DE10-Nano käyttää ohjaukseen yksinkertaista GNU/Linux-jakelua, joka ladataan aina koneen käynnistyessä. Tähän kuluu käytännössä vähemmän aikaa kuin hitaammalta televisiolta tahdistua koneen kuvasignaaliin. Wikistä löytyy helppokäyttöinen ohjelma, jolla microSD-kortti valmistellaan laitteeseen sopivaksi.

MiSTin tapaan MiSTeriä ohjataan USB-näppäimistöltä: aluksi valitaan, mikä core koneeseen halutaan ladata. Tämän jälkeen F12-näppäimen painallus avaa valikon, josta corelle syötetään tarvittavat käynnistystiedostot, esimerkiksi pelimoduulista tehty ROM-tiedosto tai kiintolevyn näköistiedosto ja käyttöjärjestelmä-ROM. Samasta valikosta voidaan säätää käytettävissä olevan muistin määrä, matkittava tv-standardi ja muita tarpeellisia asetuksia.

Aika monia avoimen lähdekoodin projekteja käyttäneenä olin yllättynyt siitä, miten helppoa MiSTerin käyttöönotto oli ja kuinka valmiilta tuotteelta se tuntui. Jos unohtaa sen, että kyseessä on kasa päällekkäin ladottuja piirilevyjä (joille toki voi vaikkapa 3D-tulostaa kotelon, malleja löytyy Thingiversestä), voisi luulla käyttävänsä jotakin kaupallista tuotetta. Epäilemättä myös MiSTistä saadut kokemukset ovat auttaneet tässä. Kaikki kokeilemani USB-peliohjaimet tunnistuivat suoraan, asetusten teko oli yksinkertaista ja muutenkin asiat yksinkertaisesti toimivat.

Hyvältä näyttää, pääosin

Muutaman illan käyttökokemukset MiSTerin kanssa toivat rakkauteen yllättävän vähän ryppyjä. Kirjoitushetkellä Minimig-core kaatuu levylle tallennettaessa, ohjainasetukset on mahdollista tallentaa hieman hassusti ja virtakytkimen puute tekee käytöstä vähän turhan… *spectrumia*, mutta vastapuolena esimerkiksi Mega Drive- ja PC Engine -implementaatiot toimivat niin hyvin, että minun oli vaikea erottaa niitä aidosta asiasta. Siis vaikka kyseessä ovat omat lempikonsolini.

Toki mukana on jokunen kummallisuuskin. Vaikka MiSTerin juuret ovat MiST-projektissa ja siitä käydään keskustelua Atari-Forum.comissa, saatavana ei ole Atari ST -corea. Suurimman osan coreista kääntänyt projekti-isäntä Sorgelig vastaa aiheesta kyselijöille, että hänestä ST ei ole kiinnostavaa rautaa, ja koska projekti on hänen harrastuksensa, hän valitsee mihin aikansa käyttää. Ketään ei toki ole kielletty tekemästä MiSTerille ST-implementaatiota, mutta toistaiseksi tekijää ei vain ole ilmaantunut.

Aika moni asia Sorgeligia tuntuu onneksi kiinnostavan, sillä kirjoitushetkellä MiSTerille on saatavana yli 30 tietokone-corea, 12 konsolia ja yli 70 peliautomaattia. Tekijöitä on tullut mukaan lisääkin, joten kaikki käännös- ja kehitystyö ei ole enää pelkästään Sorgeligin harteilla.

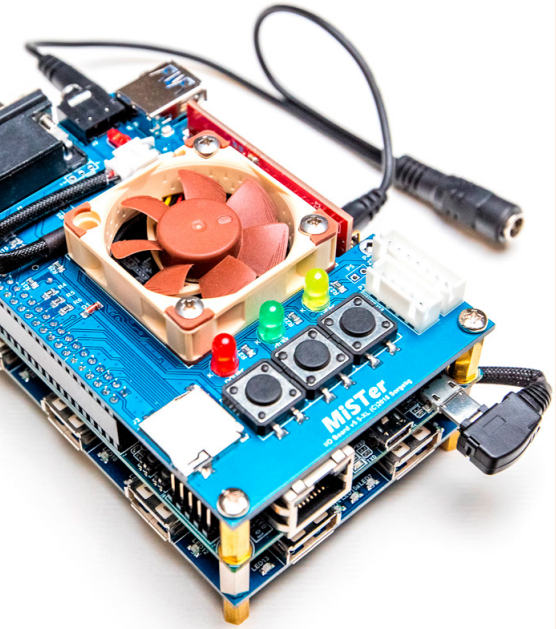
Kehitys on hyvin aktiivista, sillä suuri osa coreista päivittyy lähes jatkuvasti ja uusiakin ilmaantuu tasaisesti. Saatavana on kätevä automaattinen skripti, joka hakee uusimmat core-versiot Githubista aina pyydettäessä. Painopiste on selvästi helpommin mallinnettavissa 1980-luvun alun laitteissa, mutta toisesta päästä löytyy Amigan lisäksi mm. ao486-core, joka pyrkii mallintamaan MS-DOS-mikroa. Kuumottava on myös Japanin kotimikroylpeyttä matkiva Sharp X68000 -core, joka on vielä varsin alkutekijöissään. Monia varmasti kiinnostava Commodore 64 -core taas kärsii edelleen monista yhteensopivuusongelmista, joista Ultimate-64 ja jopa Turbo Chameleon ovat jo päässeet eroon.

Mihin vielä päästäänkään?

Mikäli kehittäjien puheita on uskominen, DE10-Nanon ja sen Altera Cyclone V:n rajoja ei ole nähty vielä likimainkaan. Suuri logiikkaelementtien määrä mahdollistaa paljon nykyistä monimutkaisempienkin kokonaisuuksien luomisen, mutta myös kehitystyö luonnollisesti vaikeutuu, kun siirrytään yhä uudemman laitteiston mallintamiseen. Esimerkiksi FPGA-PlayStationin suhteen ei kannattane pidätellä henkeään.

Siksi kannattaa muistaa kultainen sääntö ja tehdä mahdollinen hankintapäätös aina sen perusteella, mitä on tarjolla nyt, eikä luottaen mahdolliseen tulevaan kehitykseen. Nykyisellään MiSTer tarjoaa jännittävän, kompaktin ja varsin edullisen tavan matkia isoa kasaa klassikkokoneita, joten jos sydämesi sykkii kasibittimikroille, 16-bittisille pelikonsoleille ja vanhoille arcade-peleille, tuskin suuremmin petyt. Jo pelkkä FPGA-piiri erikseen ostettuna kun maksaisi enemmän kuin kokonainen DE10-Nano, ja muihin markkinoilla oleviin laitteisiin verrattuna MiSTer tarjoaa huomattavasti paremmat liitännät ja reilusti laajennusvaraa.

Linkit MiSTer-wikiin, eri komponentteihin ja testi- sekä pelivideoon löydät osoitteesta: skrolli.fi/numerot



EI NÄIN!
Amigaaargh!

Amiga-yhteisö osaa kaiken hieman muita paremmin. Myös riitelyn ja kaunaisuuden.

Teksti: Mikko Heinonen Kuva: Janina Joutsen

Tehtäköön asia heti alkuun selväksi: Amiga on myös itselleni sitä rakkainta nostalgiatietotekniikkaa, ja olen jo kauan sitten hylännyt kaikki pyrkimykseni rationalisoida siihen käyttämiäni rahamääriä. Kun nämä viisi kirjainta mainitaan, käsi hamuaa jo lompakkoa ja valmistautuu viskomaan seteleitä ruutua kohti. Ja kuten on käynyt erittäin ilmeiseksi, en ole tunteideni kanssa yksin. Amiga-fanien joukko on sankka ja uskossaan vankkumaton, ja heitä asuu joka puolella maailmaa.

Ehkä juuri tämä intohimoisuus selittää sen, että vähintään Commodoren konkurssista – josta muuten tuli juuri täyteen 25 vuotta – alkaen Amigan ympärillä on pyörinyt kaikenlaisia säätäjiä. Useimpien tarkoitusperät ovat varmasti sinällään olleet hyviä, mutta monesti lopputulos on kuitenkin ollut lihava riita.

Punaiset vastaan siniset

Commodoren kaaduttua Amigan oikeudet siirtyivät ensin saksalaiselle Escomille, joka yritti käynnistää laitevalmistusta tosissaan, sitten amerikkalaiselle Gatewaylle, joka oli kiinnostuneempi patenttisalkusta. Lopulta Commodore- ja Amiga-brändit lähtivät eri suuntiin (ks. myös Skrolli 2017.3). Omistajasta riippumatta selvää oli, että koneen tekniikka oli jo iäkästä ja käyttöjärjestelmäkin tarvitsi suuren päivityksen. Joten luonnollisesti niitä syntyi kaksi.

AmigaOS 4 on ”virallinen” jatko Commodoren AmigaOS-versioille ja toimi alunperin vain tietyissä PowerPC-suoritinta käyttävissä AmigaONE-tietokonemalleissa. Niitä valmisti ensin Eyetech, sittemmin ACube Systems ja A-EON Technology. MorphOS taas on ”Amigan kaltainen” käyttöjärjestelmä, joka toimii erilaisissa PPC-laitteissa, kuten vanhoissa Apple Macintosheissa sekä Genesi USA:n valmistamissa Pegasos-tietokoneissa.

En edes yritä vertailla näitä kahta käyttöjärjestelmää, joilla varmasti molemmilla on omat ansionsa ja vastaavasti puutteensa. Samoin jakautumisen perussyy vaikutti varmasti asianosaisista kovin tärkeältä aikoinaan. Sen sijaan katson asiaa innokkaana Amiga-harrastajana, jonka olisi pitänyt vuosituhannen vaihteen puolella osata valita puolensa taistelussa: toisella puolella aitaa olivat ”punaiset” AmigaOS 4 -kannattajat, toisella ”siniset” MorphOStelijat. Foorumeille suollettiin sivukaupalla tekstiä puolesta ja vastaan, eikä periksi annettu tuumaakaan. Ulkopuolisen silmään näytti siltä, että tuettu laitteisto ja koko projektin päämäärä olivat lopulta hyvin samanlaisia, mutta molempien tiimien piti saada tehdä työnsä yksin. Lähtökohtaisestikin pieni potentiaalinen käyttäjäjoukko jaettiin näin kahtia, ja lopputuloksena varsin moni mahdollinen harrastaja teki samoin kuin minä ja päätti nakata tuhkat koko touhulle. Vanha 68k-Amiga jäi pöydälle muistoksi menneistä ja tietojenkäsittely hoidet-

tiin jatkossa Microsoft–Apple–GNU/Linux-akselilla.

Rintamalla on tapahtunut liennytystä ja laitetukea on laajennettu molemmin puolin, mutta molemmat käyttöjärjestelmät ovat tänäkin päivänä erikseen olemassa ja myytävänä. Niiden rinnalle on ilmestynyt yksi lisääkin, tai oikeastaan jo ennen niitä: avoimen lähdekoodin AROS, joka pyörii (muun muassa) Intel-arkkitehtuurilla ja tekee jälleen kerran ne samat asiat kuin AmigaOS 4 ja MorphOS eli tarjoaa API-yhteensopivuuden vanhoille Workbench-ohjelmille ja emulaatiotukea muulle Amiga-tavaralle. Niin, ja myös klassinen AmigaOS muuten päivittyy edelleen.

Kelloportti-Jensin monet keksinnöt

Skrollissakin monesti mainittu **Jens Schönfeld** on saksalainen elektroniikkainsinööri, joka on tehnyt klassikkokoneiden harrastuksestaan pikkubisnestä kehittelemällä erilaisia laajennuksia sekä Amigalle että Commodore 64:lle. Jensin käsialaa ovat Indivision-näyttösovittimet, ACA-kiihdytinkortit, C64 Reloaded -emolevyt, X-Surf-verkkokortit ja Turbo Chameleon -FPGA-mikro, noin muutamia mainitakseni (ks. esim. Skrollit 2018.1, 2018.2). Lisänimensä ”Clockport” hän sai aikoinaan tavasta käyttää Amiga 1200:n kellovälyä mitä moninaisimpiin laajennuksiin.

Toisin kuin moni muu marginaaliraudan valmistaja, Jens on pyrkinyt myyntitoiminnassaan ammattimaisuuteen. Individual Computers on ihan oikea yritys, jolla on verotiedot kunnossa ja verkkokaupan varastosaldot pitävät kutinsa. Tuotteilla on pakkaukset ja painetut käyttöohjeet.

Schönfeldillä on kuitenkin myös äänekkäät kriitikkonsa, joiden mielestä hän antaa katteettomia lupauksia ja hylkää tuotteitaan keskeneräisinä myydäkseen niiden tilalle uusia. Erityisen pahamaineisena muistetaan Keyrah-adapteri, joka muutti klassikkomikrojen näppäimistöt USB-liitäntäiseksi emulaattorikäyttöä varten. Sitä mainostettiin pelikäyttöön sopivana, mutta käytännössä tämä oli mahdotonta syöteviiveen vuoksi. Catweasel Mk4 PCI -levyasemaohjain taas ei koskaan toiminut kunnolla edes Windows 7:ssä, uudemmista versioista puhumattakaan.

Uusin konflikti on syntynyt Schönfeldin ja Amiga-emulaattoripaketti *Amiga Foreveriä* julkaisevan Cloanton välille. Jens hankki itselleen oikeudet suosittuun P96-ohjelmistoon (aiemmin Picasso96), jolla Amigan saa tukemaan erilaisia näytönohjaimia ja suurempia näyttötarkkuuksia. Individual Computers ilmoitti olevansa ainoa virallinen paikka, josta P96:n shareware-version voi rekisteröidä. Cloanto on kuitenkin jaellut rekisteröityä versiota jo vuosikaudet *Amiga Foreverin* mukana ja on sitä mieltä, että heidän aikoinaan alkuperäisiltä tekijöiltä hankkimansa lisenssi siihen on ikuinen.

Cloanto ei suostunut Individual Computersin vaatimukseen P96:n lisenssin uusimisesta ja juttu siirtyi asianajajille, jolloin Jens veti niskaansa marttyyrin viitan. Vuotaneessa sähköpostissa hän ilmoitti lopettavansa kaiken Amiga-lisävarusteiden rakentamisen ja myynnin, jos Cloanto voittaa oikeusjutut tai ei vedä niitä pois vuoteen 2020 mennessä. Samalla Individual Computers veti taloudellisen tukensa ainakin yhdeltä merkittävältä Amiga-tapahtumalta.

Jälleen kerran foorumeilla tehtiin jako kahteen. Toisten mielestä Jens on kuitenkin tehnyt paljon hyvää ja tuonut uutta rautaa markkinoille, Cloanto taas vain istuu omistamiensa Kickstart-ROMien päällä ja rahastaa emulaattorillaan. Toiset taas pitävät P96:n ostamista lopullisena osoituksena Jensin häikäilemättömyydestä ja halusta vahingoittaa Cloantoa. Kolmansien mielestä kyseessä on jälleen yksi turha riita yli 20 vuotta vanhasta atk:sta. Tätä kirjoittaessani P96-lisenssejä ei voi ostaa Individual Computersin kaupasta.

Purrako vai eikö purra

Näppäimistöjä on kuluttanut myös Skrollissakin (2014.1, 2018.2) esitelty Vampire-kiihdytinkortti. Yksinkertaisena turbokorttina liikkeelle lähtenyt Vampire on nykyisin kokonaisratkaisu, jossa Amiga toimii lähinnä syöttölaitteena tehokkaan FPGA-piirin sisään mallinnetulle Amiga-yhteensopivalle tietokoneelle. HDMI-liitäntä parempine näyttötiloineen, SD-korttien tuki, verkkokortti ja perus-Amigoihin nähden moninkertainen suorituskyky ovat houkuttelevia yksityiskohtia. Apollo Core -kehittäjätiimi on myös ollut varsin aktiivinen ja päivittänyt ydintä usein.

Rintamalinja Vampire-keskusteluissa kulkeekin sen kohdalla, mikä sitten on kenellekin riittävä mallinnuksen tarkkuus. Vampireen luotu ”68080” on kuvitteellinen Motorola-suoritin, joka on monin verroin tehokkaampi kuin saman sarjan esikuvansa, mutta sisältää myös omanlaisiaan rajoituksia. Muistinhallintayksikkö (MMU) on pudotettu pois ”tarpeettomana” ja liukulukuyksikön (FPU) tarkkuutta on alennettu, jotta se mahtuu Vampiren FPGA:lle. Apollo-tiimin mukaan tarkoitus ei edes ole luoda täydellistä kloonia vaan uutta, kiinnostavaa Amiga-rautaa, ja nykyinen toteutus riittää pelien ja demojen ajamiseen.

Apollo-tiimi on objektiivisesti katsoen oikeassa, sillä Vampirella saa tehtyä sen mitä 99 prosenttia käyttäjistä haluaa. Hinta on murto-osan siitä, mitä kustantaisi koota edes likimain yhtä tehokas kokoonpano oikeista vintage-puolijohteista. FPGA:lla voidaan myös korjata monia arkkitehtuurin historiallisia ongelmia. Samoin objektiivisesti oikeassa ovat ne, joiden mielestä Amiga on muutakin kuin pelkkä syöttölaite FPGA-piirille: se on Motorolan suorittimiin ja erikoispiireihin perustuva klassikkomikro ja keräilykohde sinällään, eikä uusien, yhteensopimattomien ominaisuuksien lisääminen Vampireen juuri hyödytä muuta yhteisöä. Mutta koska kyseessä on Amiga, objektiivisuus lentää ikkunasta heti kun keskustelu alkaa. Joko vihaat Vampirea tai rakastat sitä, valitse puolesi.

Viimeinen taisto

Paikallinen jääkiekkojoukkue mainostaa, kuinka se ”menee tunteisiin”. Välillä tuntuu, että lätkähommatkin jäävät kakkoseksi, kun amigistit käyvät peitset tanassa taistoon. Nämä kolme esimerkkiä ovat vain pintaraapaisu siihen draamaan, jota eri palstoilla on saatu kehiteltyä. On selvää, että juuri intohimo aiheeseen on pitänyt laitteen hengissä ja mielekkäänä kehitysalustana näin pitkään valmistajan konkurssin jälkeen, mutta toisinaan kuppikuntaisuudesta on enemmän haittaa kuin hyötyä. Harvoin on vain yhtä oikeaa tapaa harrastaa, ja toisinaan taas kaikki ovat yhtä väärässä ja ainoa oikea tapa on lopettaa turha riitely.

Maksuttomat pdf-versiot yli vuoden vanhoista Skrolleista skrolli.fi:ssä: skrolli.fi/numerot

1

Minikonsolit

Tiesitkö, että...

Teksti ja kuva: Janne Sirén

5

3

4

10

11

Monet pelikoneklassikot ovat tehneet paluun enemmän tai vähemmän autenttisissa ja kutistetuissa kuorissa. Mutta tiesitkö tätä?

1 Peliohjaimeen pakattu C64 Direct-to-TV -minikonsoli (Skrolli 2018.2) on kuvan vaatimattomin, mutta samalla kunnianhimoisin minikone. Muiden käyttämien emulaattoriohjelmistojen sijaan laitetta pyörittää hakkeri **Jeri Ellsworthin** kehittämä FPGA-uudelleentoteutus Commodore 64:stä, tosin edullisempana ASIC-versiona. Harrastuneempi voi kytkeä C64 DTV:hen jopa aidon 1541-levykeaseman.

2 ZX Spectrum Vega -minikonsolin päällä oleva The Recreated ZX Spectrum (Skrolli 2018.4) on artikkelin ainoa retrokone, joka on esikuvansa kokoinen. Sen pakkaus on silti kuvan koneiden pienin. Alkuperäinen ZX Spectrum olikin todella kompakti: siinä ei ollut vakiona edes peliohjainportteja, minkä vuoksi Vega-minikin on muotoiltu muistuttamaan näppäimistöä.

3 Viime vuoden trendi olivat peliautomaattien neljännesversiot – uusia on luvassa tänäkin vuonna. Kuvassa Numskullin Pac-Man on mittakaavassa 1:4 (25 %). Arcade1Upilla taas on varttikoneperhe 3:4-koossa (75 %). Neljännes eli *quarter* viittaa lisäksi kolikkoon: Amerikan pelihalleissa pelattiin 25 sentin eli neljännesdollarin kolikoilla – yksi

per peli, joskus kaksi. Nelonen liittyy muuten myös edempänä mainittaviin THEC64 Miniin ja Nintendo Entertainment System Classiciin: niitä mahtuu esikuviensa pinta-alaan tasan neljä ja tilavuuteen tuplaten (mittakaava siis 1:2 eli 50 %).

4 Commodore 64 -mukaelma THEC64 Mini (Skrolli 2018.2) on silmänkääntötemppu: Näppäimistö ei toimi, eikä koneella ole nimi- tai logolisenssiä, mistä seurauksena ovat hämäävä nimi ja kauneusvirhe näppäimistön alakulmassa. Painokin on fuskausta, sillä laitteen sisälle on ruuvattu kolme metallinpalaa. THEC64:stä on kuitenkin lupailtu täysikokoista versiota toimivalla näppäimistöllä vielä vuonna 2019. Seuraa täällä: facebook.com/THEC64computer.

5 THEC64 Minin ROM-käyttöjärjestelmä on lisensoitu italialaiselta Cloantolta, joka omistaa Commodore- ja Amiga-oikeuksia. Kuvassa minikoneesta onkin myös italiankielinen painos. Maailmalta löytyy lisäksi esimerkiksi saksalainen THEC64 Mini -versio. Erot ovat lähinnä pakkauksessa, ohjekirjassa ja (valinnaisissa) valikoissa, mutta Amerikan markkinoinnin aikoihin myös itse koneen värisävy muuttui hieman.

6 Famicom on Japani-versio Nintendo Entertainment Systemistä (kuvassa yläpuolella). Mutta tiesitkö, että mini-Famicom – ja sen kullanvärinen mangapainos *Shōnen Jump Edition* – on suhteessa

Mega Drive Mini ei ehtinyt esikuvansa 30-vuotisjuhlaan Japanissa, koska Sega hylkäsi yhteistyön AtGamesin kanssa ja (onneksi) viimeistelee klassikkokonsolinsa uusimman painoksen itse. Meillä ysärikonsolina tunnettu Mega Drive julkaistiin Japanissa alkujaan jo vuonna 1988. Mega Drive Mini ilmestyy 19. syyskuuta 2019. (Kuva: Sega)

suurempi kuin muut Nintendon minikonsolit, noin 65 % alkuperäisestä. Ero johtunee Famicomin jo alkujaan pienestä koosta tai integroiduista miniohjaimista, joita ei haluttu kutistaa liikaa.

Toinenkin muutos löytyy: Famicomin kakkosohjaimessa oli mikrofoni, jota NES:ssä ei ollut. Esimerkiksi Zeldassa Polin Ääni -vihulainen (*Pols Voice*) kuoli tehokkaimmin mekastamalla mikrofoniin. Pelin ohjeissa Polin Äänen kerrotaan ”vihaavan kovaa ääntä”, mikä hämmensi pelaajia Japanin ulkopuolella. Vinkki hämmentää ohjeissa edelleen, nyt kaikkia – mini-Famicomissa kun ei ole mikrofonia, vain ritilä. Nintendo Switchin Famicom-ohjaimissa sen sijaan on mikrofoni.

7 AtGames on retrokonsolien tusinavalmistaja, joka on onnistunut lisensoimaan ja tuottamaan sinne päin -miniversioita lähes kaikista rakastetuista klassikkokonsoleista, paitsi Nintendosta. Hyvä niin – Nintendon kannalta. Tunnistat nämä Flashback-varoitustekstistä. Kuvassa Colecovision Varoitus, johon lipsahtivat muun muassa väärinpäin kytketyt vasen ja oikea tulitusnappi.

8 PlayStation-konsolista on useita painoksia. Esimerkiksi SCPH-1001-versiossa on RCA-medialiittimet, jotka poistuivat SCPH-5001:ssä. SCPH-5500:ssa painikkeiden tekstit vaihdettiin kuvakkeiksi PAL-versioissa ja SCPH-9000:sta poistettiin rinnakkaisportti. PlayStation Classic -retrokonsoli vastaa noin SCPH-5001-mallia, jossa on rinnakkaisportin luukku (koriste), painikkeissa tekstit ja alkuperäisohjaimet – paremmat DualShockit tulivat vakioksi vasta SCPH-7000:ssa.

9 Japanin Super Nintendo eli Super Famicom (kuvassa alapuolella) näytti Euroopan/PAL-alueen SNES:ltä, mutta oli peliyhteensopiva kulmikkaan Amerikan SNESin kanssa (yläpuolella), kunhan peliluukkua vähän viilasi. Eurooppa-SNES (keskellä) ei pelannut kummankaan näiden pelimoduuleita. Minikonsolit eivät käytä moduuleita, mutta asetelma on päinvastainen: pelivalikoima on yhteinen eurooppalaisilla ja amerikkalaisilla, Japani-minissä on osin eri pelit. Kooltaan mini-SNESit ovat muuten PlayStationin tapaan noin 55 % alkuperäisestä – eivät siis ”varttikonsoleita”.

10 Retrokonsolin tekeminen hyvin on taito- ja tahtolaji, jossa vain harva (siis Nintendo) onnistuu. Miniklassikoiden flopit, kuten Neo Geo mini ja PlayStation Classic, olivat esillä Skrollin edellisen numeron Ei näin! -palstalla. Sinua on varoitettu.

11 Sähköautovalmistaja Tesla ympäsi retrokolikkopeli Pole Positionin autoihinsa (Skrolli 2019.1), kunnes paljastui, ettei heillä ollutkaan kaikkia oikeuksia ja peli poistettiin kevättalvella. Internethuhu väitti, että Teslan versio olisi perustunut pelin eteläeurooppalaiseen bootleg-piraattiklooniin. Kuka tietää, kenties olet pelannut ”Tesla-versiota” jo kasarilla jossain Välimeren rantabaarissa…

12 Nintendon minikonsolien valmistus lopetettiin vuodenvaihteessa, mutta seuraavaa klassikkoa pukkaa: *Sonic The Hedgehog* palaa Sega Mega Drive Minin kera syksyllä. Mutta tiesitkö, että Mega Drive Minin Japani-versioon luvataan vuoden 1993 kuusipainikkeisia ohjaimia? Euroopassa on tyytyminen 1990-luvun vaihteen kolminappisten alkuperäisohjainten toisintoon. Lisää täällä: megadrivemini.sega.com.

SKROLLI 6,4 VUOTTA SITTEN

Tervetuloa Skrollin uusimmalle – vai pitäisikö sanoa vanhimmalle – palstalle, jolla aloitamme aikamatkan Skrolli-lehden historiaan! Olipa kerran synkkä ja myrskyisä yö, tarkalleen 6,4 vuotta sitten…

Tammikuussa 2013 valmisteltiin Skrollin ensimmäistä varsinaista numeroa. Se ilmestyi 25.3.2013, joten palataan siihen ensi numerossa. Ensimmäinen Skrolli-näyte oli kuitenkin julkaistu jo 20.10.2012 Wärkfest-tapahtumassa Helsingissä. Kerroimme tämän nelisivuisen, sangen kirjaimellisen omakustanteen tulostushaasteista Skrollin viisivuotishistoriikissa numerossa 2018.1.

Skrollin näytenumerossa 2012.0 oli muun muassa ensimmäinen **Mikko Heinosen** *Ei näin!* -palsta, joka on sittemmin ilmestynyt jokaisessa varsinaisessa Skrolli-numerossa, myös niissä kahdessa englanninkielisessä (*Fail!*-nimellä). Samaan ei pääse mikään muu Skrollin palsta, paitsi Pääkirjoitus, mutta sillä on ollut monta kirjoittajaa.

Olkaapa hyvät, kolme otetta Skrollista 2012.0:

"Uskomme, että Skrolli tulee jättämään pysyvän jäljen suomalaiseen tietotekniikkakulttuuriin. Mikäli suinkin kiinnostaa, tervetuloa mukaan tekemään historiaa!" – **Ville-Matias Heikkilä**, Pääkirjoitus

"QML ei ole alustariippuvainen, vaan toimii Windowsissa, työpöytä-Linuxeissa, OS X:ssä, Maemossa, MeeGossa, Symbianissa ja Windows CE:ssä. Lisäksi Android-, iOS- ja Windows Phone 8 -porttaukset ovat

tekeillä. Suurin QML:n käyttäjä on tällä hetkellä Nokian N9-puhelin. Jollan tuleva MeeGo-puhelin käyttää myös yksinomaan QML:ää käyttöliittymissään." – **Ville Ranki**, Opi QML:ää Skrollin parissa

"Etupäässä Samsungin valmistamiin laitteisiin päätynyt [Nuon-]teknologia lisäsi DVD-levyihin parempaa zoomausta ja pehmeämpää kelausta, mutta kykeni myös ihan kohtalaisten pelien pyörittämiseen. Nuonin harmiksi markkinat olivat jo kovin täynnä, ja pelihalukkaiden DVD-soittimen ostajien kotiin löysikin tiensä useammin Sonyn PlayStation 2." – Mikko Heinonen, Ei näin: CD-i, 3DO ja Nuon

Janne Sirén

Skrollin vanhat pdf-lehdet saat maksutta osoitteesta: skrolli.fi/numerot

Anssi Kolehmainen, Janne Sirén, Manu Pärssinen

SKROLLIKALENTERI

Tervetuloa kolmanteen Skrolli-kalenteriin! Palstalla kartoitamme tietokonekulttuurin kokonaiskuvaa päivämäärien ja tapahtumapaikkojen kautta. Tässä kattaus Suomessa ja suomalaisittain kiinnostavissa maailmankolkissa järjestettäviä tietokone-, peli- ja demoskenetapahtumia sekä huomionarvoisia yhteisötapahtumia.

PVM	TAPAHTUMA
Peliala	
16.7.2019	IGDA Finland Turku Hub - Picnic Time! Kupittaan puisto
7.8.2019	Kuopio HUB Gameskiviikko gathering, Malja
20.8.2019	Helsinki Hub August gathering, Ravintola Maxine
4.9.2019	Kuopio HUB Gameskiviikko gathering, Malja
10.9.2019	Helsinki Hub September gathering, Ravintola Maxine
24.9.2019	Kajaani Hub September Monthly Gathering, Rock House Kulma
Kulttuuri & yhteisöt	
5.–7.7.2019	Finncon 2019, Jyväskylä. Suomen suurin science fiction- ja fantasiakirjallisuustapahtuma järjestetään tänä vuonna Jyväskylän yliopiston Seminaarinmäen kampuksella. Finncon 2019:n kunniavieraita ovat kääntäjä Kersti Juva, professori Raine Koskimaa, kustantaja Cheryl Morgan ja kirjailija Charles Stross.
26.–28.7.2019	Ropecon 2019, Helsinki. Euroopan suurin vapaaehtoisvoimin järjestettävä pelitapahtuma, joka tuo yhteen alan harrastajat ja ammattilaiset. Ohjelmaan kuuluu pelaamisen ja peliturnausten lisäksi luentoja, esitelmiä, paneelikeskusteluja ja työpajoja alkaen historiallisista teemoista aina uusien pelien ideoimiseen ja käytännön toteuttamiseen.
15.–18.8.2019	Worldcon 2019, Dublin. Maailman pisimpään toiminut scifi-tapahtuma, joka kerää yhteen useita tuhansia asiasta kiinnostuneita harrastajia. Worldcon jakaa myös tieteiskirjallisuuden Hugo-palkinnot.
6.–9.9.2019	Tracon 2019, Tampere. Pohjoismaiden suurimpana roolipeli- ja animetapahtumana Tracon kerää Tampere-taloon noin 6 000 harrastajaa. Paikan päällä voi viettää aikaa pelaten pöytä- ja liveroolipelejä, lauta-, kortti,- miniatyyri- ja strategiapelejä, sekä kuunnella mielenkiintoisia luentoja näihin aiheisiin liittyen.
23.–24.11.2019	Tracon Hitpoint, Tampere. Perinteiseen pöytäpelaamiseen keskittyvä tapahtuma tarjoaa yleisölleen lauta-, kortti-, rooli-, ja miniatyyripelejä sekä muuta aiheeseen liittyvää ohjelmaa. Tapahtuma on oiva tilaisuus viettää aikaa pelaillen rennosti tai tosissaan ja tutustua erilaisiin peleihin ja muihin samanhenkisiin harrastajiin.
Syyskuu, 2019	Ihan ihme vehkeet II, Suomen pelimuseo, Tampere. Jatkoa viime vuoden menestystapahtumalle pelikonehistorian erikoisuuksista.
Demoscene	
1.–4.8.2019	Assembly 2019, Helsinki. Lajissaan maailman suurimpiin kuuluva demoparty ja tietokonefestivaali. Kolme vuorokautta putkeen pelaamista, demoja sekä muuta ohjelmaa. Tule tapaamaan Skrollin tekijöitä lehden ständille.
28.9.2019	Skrolli Party 2019, Tampere. Järjestyksessään kolmas Skrolli-lehden järjestämä tapaaminen ja demoparty – tällä kertaa Espoon sijaan Tampereella.
4.–6.10.2019	ZOO'19, Viialan Seurahuone, Akaa. Suomen suurin Commodore 64 -tapahtuma. Paikalta löytyy musiikkia, demoja, juomaa, sauna ja mieltäpäräyttävä 8-bittinen kokemus.
Esports & Lanit	
14.9.2019	Arctic Invitational, Helsinki. Suomen historian suurimmaksi esports-turnaukseksi itseään kutsuva tapahtuma järjestetään syyskuussa Hartwall Arenalla. Tapahtumassa on luvassa CS:GO:n sekä Elisan ja Hatchin järjestämän mobiilipelikiertueen finaalit.
24.–27.10.2019	Insomnia XXI, Pori. Mikäli hiirisormea kutkuttaa syysloman aikaan, on hyvä suunnata Satakuntaan ja mennä pelaamaan koko viikonlopuksi Insomniaan. Verkkopelitapahtumissa tunnelma on aivan eri kuin kotona nettiyhteyden yli pelatessa. Osallistuja voi kilpailla tuttujen ja tuntemattomien seurassa tietäen, että vastapelaajat löytyvät samasta suuresta tilasta.
(koko ajan)	Telia Esports Series. Uusi tulokas Suomen Esports-areenalla on Telian sarja, jossa kilpaillaan CS:GO:n sekä Arena of Valorin herruudesta. Pelejä on joka viikko, ja finaalipelit ratkaistaan elokuussa Assemblyillä.

Tapahtumajärjestäjä tai -kävijä, vinkkaa meille tapahtumistasi – myös pienempien paikkakuntien tilaisuudet ovat tervetulleita: kalenteri@skrolli.fi

SKROLLI PARTY
28.9.2019, Tampere
skrolli.fi/tapahtumat
SKROLLI PARTY

International Edition

Skrolli kävi ulkomailla

Teimme kaksi Skrolli-lehteä ja yhden näytenumeron englanniksi. Mitä jäi käteen vuoden 2016 onnistuneesta joukkorahoituskampanjasta?

Teksti ja kuvat: Janne Sirén

”Nyt se on päätetty: Skrolli levittäytyy ulkomaille”, kirjoitti **Ville-Matias Heikkilä** Skrolli 2015.4:n pääkirjoituksessa joulukuussa 2015. Kirjoitus toimi lähtölaukauksena Skrollin Indiegogo-joukkorahoituskampanjalle, joka keräsi 19 044 dollaria eli 127 % tavoitteestaan seuraavan kahden kuukauden aikana. Indiegogo-summa kasvoi lopulta jälkitilauksin 19 760 dollariin (noin 17 800 €).

Näin alkoi parin vuoden vientimatka eri alakulttuureja yhdistävälle, aika-agnostiselle tietokonekulttuurin erikoislehdelle. Olin mukana projektin toteutuksessa ja markkinoinnissa alusta loppuun – tällaisena se näyttäytyi minulle.

Joukkoistuksen huumaa

Skrollin alkuvuosina toimituksen piirissä ja lehdessäkin mainittiin toive, josko talkoolehti Skrollia voisi tehdä työkseen. Tämä oli yksi syy kokeilla, olisiko Skrollille kysyntää ulkomailla. Monille meistä kyse oli kuitenkin ihan siitä, että mielestämme teemme hyvää ja tärkeää julkaisua. Minusta teemme Suomen parasta tietokonelehteä. Tämä olisi ilo jakaa muuallekin. Skrollin tilaajakunta oli vakiintunut 2 000:n tasolle, mutta isossa maailmassa väkeä olisi paljon Suomea enemmän.

Ville-Matias alusti projektia toteamalla pääkirjoituksessaan, että ”Skrollin kaltaista lehteä ei suuressa maailmassa oikeastaan ole.” Tämä olisi sekä siunaus että kirous. Toisaalta suoria kilpailijoita ei ollut, mutta hyvät vertailukohdatkin puuttuivat. Suomessa rakkaus Skrolliin syntyi osittain siksi, että kotimainen tietokonemedia hylkäsi juurensa vuosikymmenen alussa tai paljon aikaisemmin – ja jatkoimme siitä. Seisoimme isoilla, olkatopatuilla hartioilla. Maailmanlaajuisessa mittakaavassa yhtä selkeää aukkoa ei ollut täytettävänä, vaikka paikallisesti vastaavia tarinoita oli toki muuallakin.

Joukkorahoituskampanjaa varten teimme englanninkielisen mallikannen ja käänsimme englanniksi listan Skrolli-artikkeleiden otsikoista. Teimme myös UKK:n (*FAQ*), jossa painotimme, ettemme ole retrolehti – se kun olisi väistämätön, mutta väärä vertailukohtamme. Myöhemmin teimme vielä kahdeksan sivun näytenumeron. Indiegogo-julkisuus ja Redditin Ask Me Anything -istuntomme siivittivätkin projektin maaliinsa. Reddit-julkisuus kaatoi jopa skrolli.fi:n hetkeksi.

Skrolli International Edition 2016.1E ilmestyi 18.4.2016. Tuhannen lehden painoksesta yli puolet lähti suoraan maailmalle. Irtonumeromyyntiin ja markkinointikäyttöön jäi sopivasti: tätä lukiessasi Skrolli-kauppa vetelee numeron osalta viimeisiään, joten pidä kiirettä, jos kiinnostaa. Kiitoksena rahoitustavoitteen ylittymisestä lisäsimme käännösartikkelien oheen Illuminatus-jutun ja tekemämme pelattavan demon pelistä virtuaaliselle kansilevykkeelle (ks. Skrolli 2019.1). Muropaketti uutisoi: ”Illuminatus ei hellitä – Skrolli vie legendaarisen aprillipilan ulkomaille.”

Yksi joukkorahoittaja Yhdysvalloista ehti kysellä ”kadonneen” fyysisen kansilevykkeen perään, ennen kuin ymmärsi vinkistämme kokeilla lehden kanteen painettua QR-koodia. Oikean levykkeen lisäämisellä kanteen kyllä leikiteltiin toimituksessakin, mutta sitä ei koskaan harkittu vakavasti.

Toinen kerta toden sanoo

Skrollin ensimmäinen kansainvälinen numero ei ollut mikään viraalihitti, eikä sillä ammattitoimitusta pystytettäisi. Lehti ylitti kuitenkin vaatimattomat kaupalliset minimitavoitteensa, joten päätimme kokeilla seuraavana vuonna uudestaan. Skrolli 2017.1E kasattiin ilman joukkorahoitusta ja markkinoinnissa luotettiin sosiaaliseen mediaan. Numero ilmestyi 10.8.2017 Assembly Summer -tapahtumassa Helsingissä ja postitettiin ennakkotilaajille tämän jälkeen.

Ensinumeron tapaan kyseessä oli kattava kokoelma suomenkielisten Skrollien artikkeleita, sopivasti päivitettynä ja käännettynä englanniksi oman ammattikääntäjämme **Mikko Heinosen** toimesta. Amerikkalainen **Michael Harlan Lyman** vastasi natiivioikoluvusta tälläkin kertaa. Vastaanotto oli lämmin, kuten hollantilaisen asi-

akkaan palautteesta voi päätellä: ”They arrived! And holy frigging crap, this magazine is fantastic! Like you folks made it just for me!” Jos saatua palautetta on uskominen, he jotka lehden löysivät, tuntuivat lähes poikkeuksetta pitävän siitä.

Olennainen osa Skrolli International Editionin jatkoa oli Skrollin digilehtiprojekti, joka valmistui myös vuodeksi 2017 (ks. Skrolli 2017.1). Siinä missä Skrolli 2016.1E julkaistiin alkujaan vain paperi- ja pdf-lehtinä, uusi 2017.1E tuli myyntiin myös Applen ja Googlen sovelluskauppoihin Skrolli-sovelluksen muodossa. Halusimme nähdä, olisiko digilehdestä hyötyä kansainvälisessä jakelussa, joten pistimme englanninkieliselle digilehdelle vieläpä huomattavasti paperilehteä edullisemman hinnan.

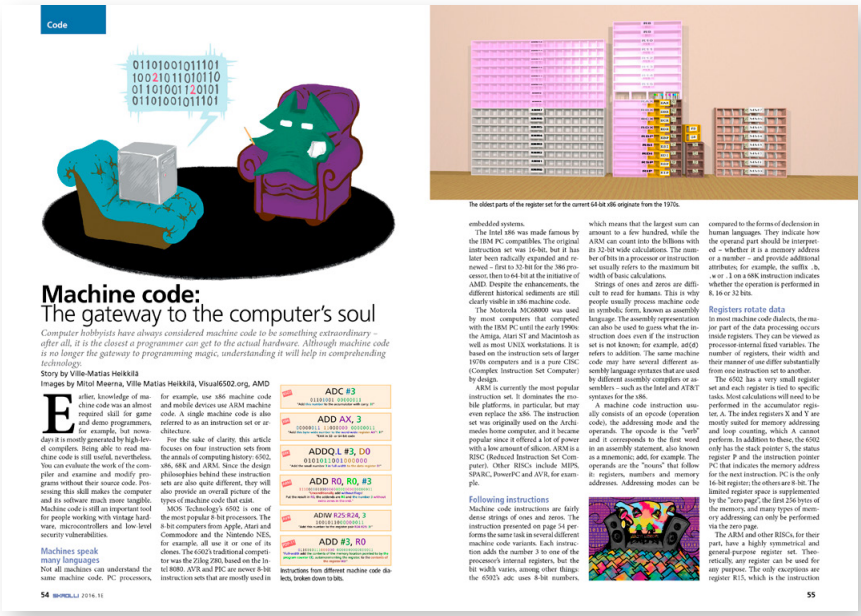
Kaupallisesti Skrolli 2017.1E:n menekki oli kuitenkin vaatimatonta sekä paperilla että digitaalisena. Paperilehteä on myyty verkkokaupassa noin kaksisataa kappaletta. Lisäksi lehtiä on myyty joitakin kymmeniä eri tapahtumissa Suomessa. Digitaalisen julkaisun tarina on suhteellisesti parempi kuin Skrollin suomenkielisellä puolella, jossa digilehtisovellus ja pdf-lehti myyvät vain sadasosan paperilehden määrästä – englanninkielistä on myyty parikymmentä eli sentään kymmenesosa. Verkossa lukijat tuppaavat suosimaan webiä tai sosiaalista mediaa, eivät näköislehtiä…

Skrolli 2017.1E:n tarina jatkuu markkinointikäytössä. Tammikuussa 2019 tietoturvaguru **Donnie Werner**, jonka haastattelu on numerossa, auttoi jakamaan allekirjoittamiaan lehtiä tapahtumavierailla Disobey 2019 -tapahtumassa Helsingissä. Englanninkielinen lehti onkin ennen kaikkea käyntikorttimme maailmalle. Niitä on lähtenyt säännöllisesti esimerkiksi kansainvälisille avustajille ja haastateltaville, viimeksi Yhdysvaltoihin, Saksaan, Kanadaan ja Australiaan.

Markkinoinnin vaikeus

Skrolli International Edition -projekti osoitti, että englanninkielisen lehden tuottaminen onnistuu Skrollilta siinä missä suomenkielisenkin. Lehdet ovat minusta mainioita. Niiden myynti ja jakelu on kyllä sitten asia erikseen. Yksinään paperin postittaminen maailmalle on hidasta, joten jaettu kokemus lehden saapumisesta (#paraspäivä) jää kokematta. Näin myös Skrollin paras markkinointipäivä puuttuu.

Ensimmäisen Skrolli International Editionin kannen kehitys joukkorahoituskampanjan aikana: esimerkkikuva (vas.), digitaalinen näytenumero ja lopullinen lehti. Kannen pohjana oli Tomi Väisäsen kehuja saanut Skrolli 2015.3:n kansikuva.

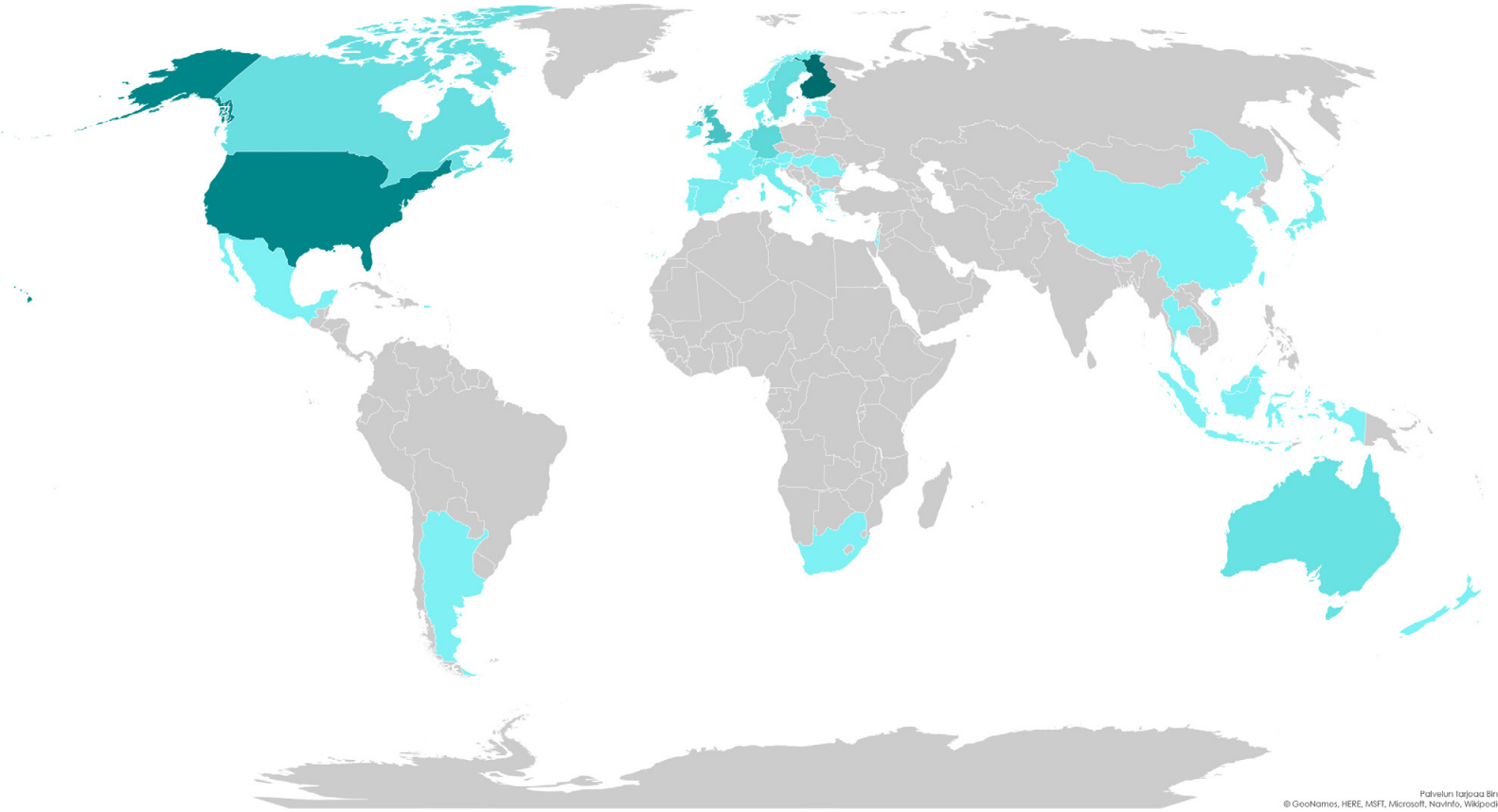


Skrollia englanniksi. Skrolli-trolli ja Ville-Matias Heikkilä johdattelivat ykkösnumeron lukijoita konekielen saloihin. Kuten suurin osa Skrolli International Editionin sisällöstä, tämäkin artikkeli on alkujaan julkaistu suomenkielisessä Skrollissa (2015.2).

Logistiikan järjestely on muutenkin kallista ja vaikeaa. Skrolli 2016.1E oli suojattu sellofaanilla, mutta sekään ei ollut aina tarpeeksi sille silppurille, jota kansainväliseksi postiksikin kutsutaan. Kaikkialla ei myöskään oltu totuttu osoitteen painamiseen lehden taakse ja jonkin keräilykappaleen vaihdoimmekin tästä syystä uuteen puhtoiseen. Skrolli 2017.1E:n kohdalla Postin tarjoamat esimaksetut postituskuoret osoittautuivat lopulta varsin kohtuuhintaiseksi ratkaisuksi, mutta hidasta se postin kulku on edelleen.

Tuotteemme osoittautui myös vaikeaksi markkinoitavaksi. Indiegogonostetta ja Redditin AMAa lukuun ottamatta mikään useista sosiaalisen median markkinointitempauksistamme ei tuottanut suurta vastakaikua, eivät edes maksetut mainokset. Pieniä puroja tuli kyllä monesta suunnasta, mutta yksittäiset innokkaat tilaajat eivät vielä tilaajakantaa tee.

Kuriositeettina **Wallun** piirtämä Skrolli 2015.2:n kansi houkutteli eteläkorealaista Facebook-yleisöä. Sieltä saatiin satoja Facebook-seuraajia, mutta kaupallista arvoa tällä ei juurikaan ollut. Etelä-Koreasta tuli joukkorahoituskampanjalle 11. eniten vierailukertoja – enemmän kuin vaikkapa Norjasta tai Tanskasta – mutta vain yksi tilaaja. Facebookin algoritmi ei edes myöhemmin tuntunut näyttävän sisältöjämme näille seuraajille. Myös Japania yritettiin houkutella sarjakuvilla, ”paremmin” tuloksin: kaksi joukkorahoittajaa ja myöhemmin kaksi lisätilaajaa. Veikkaan kielimuuria.



Skrolli 2016.1E:n ja 2017.1E:n myynti Indiegogossa ja Skrolli-kaupassa väritettynä kartalle: 42 maata, joista eniten tilaajia Suomesta (277), USA:sta (199), Isosta-Britanniasta (75), Saksasta (41), Ruotsista (33), Kanadasta (29), Australiasta (25) ja Hollannista (20). Vähintään 10 tilausta saatiin myös Sveitsistä, Norjasta, Irlannista, Tanskasta, Italiasta, Belgiasta ja Espanjasta. Muutamia tilauksia tuli esimerkiksi Japanista, Singaporesta ja Puerto Ricosta (mukana USA:n luvussa) sekä yksittäiset tilaukset muun muassa Kiinasta, Etelä-Koreasta, Seychelleiltä ja Israelista. Itä-Euroopan suoranainen poissaolo sen sijaan yllätti.

Kielestä riippumatta Skrollia on myös hieman vaikea määritellä kansainväliselle yleisölle, ja siksi yleisöä on vaikea tavoittaa. Epäkaupallinen, kaikkea kaikille -tyyppinen lehti – sekä aikajänteeltään että asiasisällöltään – on outolintu maailmassa, joka pursuaa kaupallisia tuotetestimedioita, erikoistuneita brändilehtiä ja häpeilemättömiä retrovedätyksiä. Kun tähän yhdistetään vielä arkaaiseksi koettu paperimedia, on myyjän haaste ainakin kohdallaan.

Mitä tekisin toisin?

Selkein muutos, jonka tekisin mahdollisella uusintakierroksella, olisi jatkon sitominen joukkorahoituskampanjaan. Tätä ei tehty vuonna 2016, koska oli epävarmaa, pystyisikö toimitus sitoutumaan useampaan kansainväliseen lehteen. Olimme myös kohteliaita ja lupasimme olla häiritsemättä rahoittajiamme sähköposteilla tulevaisuudessa, elleivät he erikseen kirjaisi itseään sähköpostilistallemme. Niinpä vuoden 2017 lehti ei täysin hyötynyt edellisvuoden nosteesta. Sähköpostilistallemme kirjautui 229 henkilöä, ja jotakuinkin sama määrä myytiin 2017.1E:tä.

Totesimme, ettei muutenkaan kannata aliarvioida joukkorahoituskampanjan mainosarvoa. Kampanja osoittautui arvokkaaksi markkinointivälineeksi, joka villitsee seuraajia ja puskaradiota. Tämän puute näkyi vuoden 2017 lehden myyntimäärissä. Redditin AMAa olisi voinut myös kokeilla uudestaan, mutta toimituksessa pohdimme, olisiko meillä uutta sanottavaa. Kokonaisuutena kansanvälinen yleisö vaatisi ehkä isompia tarinoita, kuin olimme tällä kertaa valmiita kertomaan.

Jatkuvuus olisi tärkeää. En tiedä kuinka suosittu suomenkielinenkään Skrolli olisi, jos ensinumero 2013.1 olisi ollut kertajulkaisu. Osa aikakauslehden lumoa on siinä, että se kolahtaa postiluukusta säännöllisesti. Jos sitoutuneita tekijöitä löytyisi, kannattaisi ehkä joukkorahoittaa kerralla englanninkielinen vuosikerta tai kaksi. Jatkuvuus voisi rohkaista jopa digilehtisovelluksen asentamiseen. Apple News+:n kaltaiset lehtien koostepalvelut voisivat myös auttaa.

Suurin haaste kuitenkin jäisi. Vaikka Skrollin sisältö kääntyy helposti englanniksi, itse *Skrolli* ei käänny. Enkä tarkoita lehden nimeä, suomenkielinen nimi ei ole vaikuttanut ongelmalta. Mutta kansainvälisellä yleisöllä ei ole suoraan samaa yhteistä tarttumapintaa kuin suomenkielisen tekijä- ja tilaajakuntamme ytimellä, joka luki aikanaan MikroBittiä, Printtiä, C-lehteä ja kumppaneita, ja joilla Skrolli yhdistyy paikalliseen jatkumoon. Englanninkielinenkin Skrolli myi parhaiten Suomessa.

Skrolli ei ole retrolehti, mutta se on retro lehti. Ilman valmista tunnesidettä lentoonlähtö on vaikeampaa. Mainostimme Skrollia aikanaan Suomen televisiossa mainoslauseella ”Muistatko vielä, kuinka tietokonelehdet olivat ennen upeita?” Viittaus on Suomessa varsin yksiselitteinen. Käytimme samaa ideaa englanninkielisessä verkkomainoksessa: ”Do you still remember how computing magazines used to be great?” Kansainväliselle yleisölle kysymys tarkoittaakin sitten kovin montaa eri asiaa, maasta riippuen. Jotain hukkuu käännöksessä.

Jos sinulla on englantia lukevia tuttavia, vinkkaa tai lahjo ihmeessä Skrolli International Editionilla. Tämä laatujournalismi ei vanhene.

Skrolli International Edition 2016.1E ja 2017.1E ovat myynnissä alennettuun hintaan Skrolli-kaupassa, paperilla ja diginä: tilaaskrolli.fi

Tekoäly kirjoitti taas

Kokeilin Skrolli 2019.1:n artikkelin *Neuroverkko kirjoitti artikkelin* innoittamana, voisinko generoida parempilaatuista Skrolli-puppua AWD-LSTM-mallilla. Mallin toiminta eroaa textgenrnn-ohjelmasta siten, että ensinnäkään se ei perustu merkkeihin vaan ”sanakkeisiin”, jotka ovat useammasta merkistä koostuvia sanan osia. Tyypillisesti sanan vartalo on yksi sanake ja taivutuspääte toinen, mutta jotkut sanat koostuvat vain yhdestä sanakkeesta ja jotkut useammasta kuin kahdesta. Malli ei ymmärrä myöskään isojen ja pienten kirjainten eroa. Uskoin, että näiden seikkojen vuoksi mallin tarvitsee käyttää vähemmän kapasiteettia suomen morfologian oppimiseen ja se voi oppia kieliopin paremmin.

Toinen ero textgenrnn-malliin on, että mallia ei ole koulutettu vain Skrolli-aineistolla, vaan se on ensin koulutettu noin gigatavulla sekalaista suomenkielistä tekstiä ja vasta sen jälkeen se on hienosäädetty Skrolli-aineistolla (toim. huom. jaoimme Skrolli 2019.1:ssä käytetyn aineiston kirjoittajalle). Hienosäätämisen ansiosta malli voi oppia luomaan kieliopillisesti oikeaa tekstiä pienestäkin aineistosta, sillä perustieto kielestä tulee pohjalla olevasta mallista.

Ohessa on esimerkki mallin generoimasta tekstistä. Alussa on ihmisen kirjoittama kehote, tässä tapauksessa ”pelistä tuli suosittu”, jonka jälkeen malli ennustaa sanakkeita.

pelistä tuli suosittu. yhdessä vaiheessa pelien aikakausi alkoi, kun amigalle tuli ensimmäiset amigaos-yhteensopivat pelit. genren konsolit tehtiin jo aivan samoihin aikoihin, ja ne tulivat markkinoille vuonna 1983. pelin ensi-ilta oli vuonna 1976. ultimate teki jatko-osaa html-moduulin the hobbit -pelikonsolille. peli julkaistiin vuonna 1986. nintendon pelihistorian varhaiset pelit olivat vain syvyyttä, mutta konsoli oli pysynyt pitkään hengissä. vaikka pelikonsolit olivatkin nyt todella viehättävät, ne olivat siinä tapauksessakin sitä, että game boyn pelien tapaan pelien historia oli muuttunut. moni halusi päästä alkuun, eikä pelimakuista ollut koskaan tarpeeksi paljon mitään tuttua.

Lisää esimerkkejä löytyy osoitteesta: github.com/fergusq/lstm-testi/blob/master/parhaat/skrolli.md.

Iikka ”fergusq” Hauhio

Tuli vaan mieleen...

Kaivakaapa asiaa tästä miehestä ja kirjoittakaa juttu: **Seymour Cray**, ”the father of supercomputing”. Hän on yksi 1960-luvun jätkistä, jonka ansiosta nykyisin ollaan tässä missä ollaan.

Myöskin CDC, Control Data Corporation, ja **William ”Bill” Norris** olisivat muistamisen arvoisia – samoin CDC Cyber -tietokoneet, jotka 60-, 70- ja 80-luvuilla pöllyttivät kaikki muut.

Nimim. ”Jo 60-luvulla rakennettiin 64-bittisiä tietokoneita, kuten myös 30-luvulla V12-moottoreita”

Kiitos erinomaisista juttuideoista! Tässä hengessä suosittelemme artikkeleitamme Supertietokoneet – Laskentatehon linnakkeet (Skrolli 2014.1) ja Toisenlaiset tietokoneet (Skrolli 2016.1), pdf-lehdet maksutta: skrolli.fi/numerot. Jälkimmäistä numeroa myydään vielä paperillakin: skrolli.fi/kauppa. Otamme mielellämme vastaan lisää sekä kirjoituksia että kirjoittajakandidaatteja tältä(kin) tietotekniikan aikakaudelta osoitteessa: toimitus@skrolli.fi.

Skrollin toimitus

Ykkösnumeron satoa

Kysyimme maalis-huhtikuussa lukijoiltamme palautetta Skrollin edellisestä numerosta 2019.1. Kun Skrollin Facebookissa, Twitterissä ja Instagramissa järjestettyjen äänestysten äänet laskettiin yhteen, 84 % piti numeroa parempana kuin edellistä 2018.4-lehteä. 16 % piti 2018.4-numeroa parempana.

Myös Skrollin numeroiden 2018.3–2019.1 aikana tapahtunut lehden ulkoasu-uudistus sai kannatusta. Yhteensä 76 % piti lehden uutta ulkoasua vanhaa parempana. 24 % piti vanhasta enemmän. Tosin Twitterissä, joka muista someista poiketen sallii kolmannen äänestysvaihtoehdon, 35 % totesi myös, ettei heillä ollut suosikkia vaihtoehdoista.

Sisältöpuolella, jos pitäisi valita, yhteensä 56 % piti Skrollin moderneja artikkeleita historiallista sisältöä tärkeämpänä. Vastaavasti 44 %:lle vanha sisältö oli uutta tärkeämpää. Molempi parempi -tunnelmat hallitsivat kuitenkin kirjallisissa kommenteissa. Jatkamme tutulla linjalla…

Saimme myös dataa postilogistiikasta. Kustannusten kanssa kipuileva

Suomen Posti pyysi luvan jakaa osan Skrolleista perjantain sijaan jo torstaina. Äänestystemme mukaan 66 % sai lehtensä torstaina. Lopuista pääosalle lehti vaikutti tulleen perjantaina. Joillakin sen saapuminen meni harmillisesti viikonlopun ylitse.

Poimintoja sosiaalisessa mediassa saamastamme kirjallisesta palautteesta:

”Skrolli 2019.1:ssä uudenuusi ja muinainen kama sulautuivat yhteen kerrassaan herkullisen yhtenäiseksi kokonaisuudeksi.”

”Tärkeämpää kuin aikakausi on Skrollin asenne, johon kuuluu mm. tee-se-itse-henki, luovuus ja tietty kriittisyys.”

”Skrollin ei ehkä kannattaisi yrittää elää rinnakkaistodellisuudessa, jossa tarvittaisiin Skrolli kertomaan mainstreamin nykyteknologian hyödyntämisestä (esim. AWS).”

”En liioittele, jos sanon että Skrollissa 2019.1 oli paras suomenkielinen tekoälyjuttu mitä olen lukenut.”

”Itselleni tekoälyteemaa oli ehkä vähän turhan monta aukeamaa, vaikka se olikin oikein mielenkiintoista luettavaa!”

”Olenko ainoa, joka on sitä mieltä, että Skrollin taso on parantunut paranemistaan koko olemassaolonsa ajan? Ja ne ensimmäisetkään eivät olleet huonoja, päin vastoin.”

”Luen työkseni ja huvikseni koko ajan, mutta tämä aviisi on kyllä ihan omassa luokassaan. Intohimoa, rosoa, sokean piripäisen hillerin konformistisuutta.”

Kiitos palautteesta!

Janne Sirén, Skrollin toimituspäällikkö

JOKSTORIAA

osa IX: Grendel-hifiä ja salanimiä

Teksti ja kuvat: Jukka O. Kauppinen

Kaverieni keskuudessa elää käsite Grendel-hifi, jolla viitataan etenkin äänentoiston vaatimattomuuteen. Vaikka kiistänkin sen yleispätevyyden kaikkeen, niin totta on, että hyödynnän tarkoituksenmukaisesti toimivaa tekniikkaa niin pitkään kuin mahdollista.

Olohuoneessani on edelleen vuonna 1985 ostamani stereosarja. Hienosta stereokaapista tosin puuttuu nykyään tuplakasettidekki, joka oli alkujaan tärkeä syy tämän laitteen ostamiselle. Tupladekin kasettipäiden asetukset näet toimivat täydellisesti kuusnelosen pelikasetteja kopioidessa. Vinyylisoitin on yhä käytössä, vaikkakin muutamaa hihnaa ja neulaa uudempana. Vahvarin ulkoiset liitännät ovat vähissä, mutta äänisignaalin jakajalla siihen saa pelikonsolien, dvd/cd-soittimen ja minkä vain muun vekottimen äänet.

Täytyy muuten sanoa, että olihan silloin komeaa luukuttaa kuusnelosmusaa isoista stereoista. Näissä kun on edelleen hyvät ja niin isot kaapit, että kyllä muuten lähtee. Tietysti nauhoittelin parhaita biisejä myös kasetille. Niitäkin taitaa olla jokunen yhä hyllyssä – SIDien parhaita.

Televisioita työ- ja olohuoneessani on nähty vuosikymmenten mittaan muutamakin. Pari viimeistä ostin yksinomaan teknisten ominaisuuksien ja liitäntöjen perusteella niin, että ne olisivat mahdollisimman hyviä pelikäytössä. Niiden edeltäjä oli todellinen työn sankari, joka ehti palvella sukuamme 29 vuotta ja 11 kuukautta.

Laite oli suorastaan historiallinen, sillä se oli Ylä-Savon ensimmäinen väritelevisio. Ihan totta. Jossain vaiheessa tämä upea reliikki siirtyi minulle, vaikka se jäikin vähälle käytölle. En katsonut tv:tä, koska tietokoneen näpyttely oli kivempaa, vaikka *Babylon 5* ja *Star Trek* murensivatkin vastarintaani. Sitten eräänä päivänä päätin, että nyt meikän on aika siirtyä konsoliaikakauteen, ja viikon sisään talouteeni saapuivat PlayStation 2, Xbox, GameCube ja Game Boy Advance.

MIKRO SYKE

SYKE sivu 8

Videopelien suosio laskee

Vierailu RAY:ssä

Leike Keskisuomalaisen Syke-liitteessä julkaistusta artikkelista. Kävin Petteri Vuorelan kanssa Kuopiossa tutustumassa Raha-automaattiyhdistyksen toimintaan.

Sitten tajusin, että reliikkini liitännät olivat vajavaiset. Siinä oli vain RF-antenniliitin.

Siinä saikin hetken miettiä, että miten minä nämä konsolit kytken. Ratkaisu oli konsolien SCART-videoulostulo. Liitin kaikki konsolit scart-jakajaan, josta signaali siirtyi edelleen videonauhurin kautta RF:nä televisioon. Tadaa! Ratkaisun yllättävä etu oli se, että pystyin tallentamaan konsolipelien kuvaa VHS/SVHS-nauhalle. Se oli eräässä vaiheessa yllättävän hyödyllistä myös kaupallisessa mielessä. Sanotaan näin, että olette todennäköisesti nähneet tv:n peliohjelmissa meikäläisen tilauksesta pelaamaa ja nauhoittamaa pelikuvaa.

Lisäbonus tuli siitä, että televisiovanhukseni kuva oli niin pehmeä, että kaikkien konsolien grafiikka sai automaattisen antialiasoinnin. Niin mitkä terävät polygonit?

Vaan aikansa kutakin, ja eräänä päivänä tv-vanhukseni päästi toimintasavut pihalle. Käytin sitä ihan vain periaatteen vuoksi näytillä korjaajalla, joka päätään pudistellen totesi, ettei tähän saa mistään varaosia. Harkitsin reliikin kannibalisointia niin, että modaisin rungosta ainutlaatuisen rottahäkin. Rotta-tv:ssä olisi aina hauskaa ja vipeltävää ohjelmaa! Mutta tilanpuute ei sallinut laitteen säästämistä. Niinpä tein tv-tekniikassa 30 vuoden aikahypyn, ja 100 hertsillä polygonit saivatkin aivan uutta terävyyttä. Olihan se hyvä tv, mutta tylsä, ja vieraiden ”mikä ihmeen telkkari sinulla on?” -keskustelunavaukset katosivat.

Aloitteleva toimittaja

JOKstorian kuudennessa osassa muistelin kaverini **Petterin** Atari 800XL -mikroa ja sen äärimmäisen äänekästä tulostinta. Oli muuten vekotin – tulostaessa ei voinut olla samassa huoneessa ja mekastus kuului talon läpi ulko-ovelle. Mutta olihan se niin hieno laite, että perhe päätti hankkia sellaisen minullekin. Hyvän tulostimen sainkin, sillä Star NL10 palveli minua vielä pitkälle Amiga- ja PC-aikakauteen. Nykyään jo vähän harmittaa, että luovuin siitä, mutta harval-

lapa meistä on loputtomasti varastotilaa.

Sitä ennen kirjoitin juttuja ihan perinteisesti mekaanisella kirjoituskoneella. Tein ensin muistiinpanoja ja laadin muistivihkoon jonkinlaisen jutturungon, joka muovaantui varsinaisen kirjoituksen aikana päässä valmiiksi artikkeliksi. Kirjoituskoneella naputtaminen oli siitä hupaisaa, että sehän oli aivan armotonta. Kun kirjain lyötiin paperille, niin siinä se seisoi. Yksittäiset virheet pystyi korjaamaan korjauslakalla niin, että paperi kierrettiin pois kirjasinvasaran tieltä, lakattiin peittoon, annettiin kuivua ja sitten lyötiin oikea kirjain tilalle. Se oli hidasta ja työlästä, joten oli parempi varoa virheitä. Vedoksen kanssa ei toki ollut niin tarkkaa, mutta lopullinen juttu piti saada paperille oikein.

Kirjoituskoneella kirjoittaminen oli siis haastavaa, etenkin kun meidän koneemme oli jo silloin ikivanha mekaaninen rumilus, jota sai takoa ihan tosissaan. Mutta mikäs siinä, nuoruuden innolla ja energialla. Kirjoista, kirjoittamisesta ja tarinoinnista aina tykänneenä koin konekirjoittamisenkin mieluisaksi toiminnaksi, etenkin kun pystyin yhdistämään sen tietokoneen näpyttelyyn ja muuhun hyödylliseen.

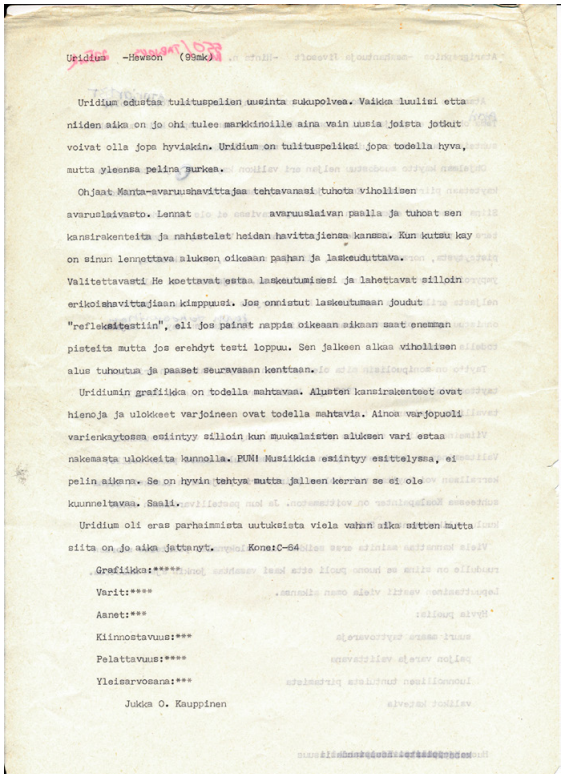
Ensimmäinen juttunihan oli (Skrolli 2018.3, JOKstoriaa 6) Keskisuomalainen-lehden nuorisoliite *Sykkeessä* 1986 julkaistu peliarvostelu *Mandragore*-pelistä. Sen ilmestyminen aukaisi sisäiset tulvaportit, minkä myötä ryhdyin kirjoittamaan Sykkeeseen todella aktiivisesti. Peliarvostelujen lisäksi innostuin kirjoittamaan myös muista kulttuurisista aiheista ja ilmiöistä. Toimituksen ystävällinen väki suhtautui oman lukija-alueensa ulkopuolelta tulleeseen kirjoittajaan kannustavasti ja opastavasti, joten koin touhun todella motivoivaksi ja miellyttäväksi. Kiitos teille!

Halu kirjoittaa kasvoi, joten otin yhteyksiä moneen eri suuntaan. Muutaman vuoden sisään kytköksiä syntyi myös omaan maakuntalehteemme *Savon Sanomiin* sekä paikallisempiin julkaisuihin *Salmetar* ja *Iisalmen Sanomat*. Lehtiin pääseminen oli varsin helppoa, sillä kirjoittajia ei juuri tunkenut joka ovesta ja ikkunasta. Lähtökohtaisesti pystyin tarjoamaan juttuja aiheista, joille ei ollut paljon kirjoittajia, ja kaiketi osasin kirjoittaakin jollain tasolla luettavaa tekstiä. Suurin kynnys oli tietysti ensimmäinen yhteydenotto. Kirje oli liian etäinen eikä sähköpostia vielä ollut, joten yleensä otin puhelimen kouraan, nieleskelin viisi minuuttia paniikissa ja aloitin numeron *veivaamisen* useamman kerran alusta, kun pelkotilat pakottivat lyömään luurin kiinni. Sitten jännäsin puhelimessa aikani ja lähetin luvatun näytejutun perään kirjeenä – eli alkuun kirjoituskoneella hakattuna tekstinä.

Lehtikavalkadin laajetessa täytyi miettiä myös omaa identiteettiään. Kyllähän te tiedätte, että ihmiset voivat olla ikäviä. Ensimmäiset Savon Sanomissa ilmestyneet jutut tulivat myös luokkatovereiden tietoon, mikä aiheutti tietenkin kettuilua ja naljailua. Että tuo tyyppi kehtasikin kirjoittaa lehtiin, miten tyhmää!

Ei siis ihme, että laajentaessani mediakenttääni kotikonnuille päädyin käyttämään salanimeä, jopa useita. Vanhoja juttuja selatessani löysin viisi eri nimimerkkiä tai salanimeä, joita käytin paikallislehdissä: kahta käytin puhtaasti tietokonejuttuihin, yhtä musajuttuihin ja yhtä sarjakuva- ja kirjajuttuihin. Bonuksena pystyin osallistumaan koulun käytävillä omia juttujani koskeviin keskusteluihin. Erästä niistä käytän muuten edelleenkin, mutta tohtinen paljastaa, että yksi käytöstä poistuneista oli KaJ eli Kakka Juippinen.

Tämä meni nyt melkoiseksi peilikuvan kehumiseksi. Jatketaan sitä seuraavassa Skrollissa tutkimalla peli- ja viihdejournalismin tilaa vuosina 1986–1987.



Uridium -Hewson (99mk)

Uridium edustaa tulituspelien uusinta sukupolvea. Vaikka luulisi etta niiden aika on jo ohi tulee markkinoille aina vain uusia joista jotkut voivat olla jopa hyviakin. Uridium on tulituspeliksi jopa todella hyva, mutta yleensa pelina purkea.

Ohjaat Manta-avaruushavittajaa tehtavanasi tuhota vihollisen avaruslaivasto. Lennat avaruuslaivan paalla ja tuhoat sen kansirakenteita ja rahistelet heidan havittajiensa kanssa. Kun kutsu kay on sinun lennettava aluksen oikeaan paahan ja laskeuduttava. Valitettavasti He koettavat estaa laskeutumisesi ja lahettavat silloin erikoishavittajiaan kimppuusi. Jos onnistut laskeutumaan joudut "reflekseitestiin", eli jos painat nappia oikeaan aikaan saat enemman pisteita mutta jos erehdyt testi loppuu. Sen jalkeen alkaa vihollisen alus tuhoutua ja paaset seuravaaan kenttaan.

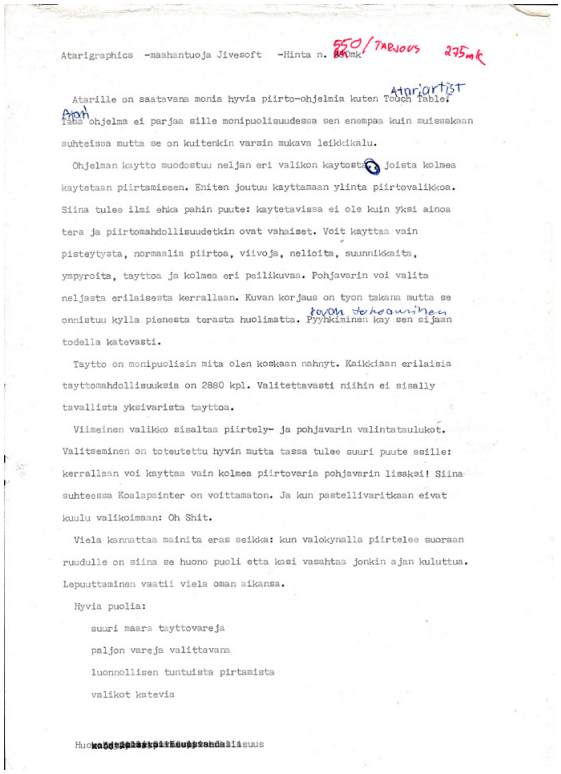
Uridiumin grafiikka on todella mahtavaa. Alusten kansirakenteet ovat hienoja ja ulokkeet varjoineen ovat todella mahtavia. Ainoa varjopuoli varienkaytossa esiintyy silloin kun muukalaisten aluksen vari estaa nakemasta ulokkeita kunnolla. PUH! Musiikkia esiintyy esittelyssa, ei pelin aikana. Se on hyvin tehtya mutta jalleen kerran se ei ole kuunneltavaa. Saali.

Uridium oli eras parhaimmista uutuuksista viela vahan aika sitten mutta siita on jo aika jattanyt. Kone:C-64

Grafiikka:*****
Varit:****
Aanet:***
Kiinnostavuus:***
Pelattavuus:****
Yleisarvosana:***

Jukka O. Kauppinen

Uridium-räiskintäpelin arvostelu on sisällöltään kiintoisan ristiriitainen. Skandien puuttuminen vihjaa, että se on tietokoneella tehdyn jutun vedosversio.



Atarigraphics -maahantuoja Jivesoft -Hinta n. 80mk

Ataarille on saatavana monia hyvia piirto-ohjelmia kuten Touch Table, tama ohjelma ei parjaa sille monipuolisuudessa sen enempaa kuin muissakaan suhteissa mutta se on kuiterkin varsin mukava leikkikalu.

Ohjelman kaytto muodostuu neljan eri valikon kaytosta joista kolmea kaytetaan piirtamiseen. Eniten joutuu kayttamaan ylinta piirtovalikkoa. Siina tulee ilmi ehka pahin puute: kaytettavissa ei ole kuin yksi ainoa tera ja piirtomahdollisuudetkin ovat vahaiset. Voit kayttaa vain pisteytysta, normaalia piirtoa, viivoja, nelioita, suunnikkaita, ympyroita, tayttoa ja kolmea eri peilikuvaa. Pohjavarin voi valita neljasta erilaisesta kerrallaan. Kuvan korjaus on tyon takana mutta se onnistuu kylla pienesta terasta huolimatta. Pyyhkiminen kay sen eijaan todella katevasti.

Taytto on monipuolisin mita olen koskaan nahnyt. Kaikkiaan erilaisia tayttomahdollisuuksia on 2880 kpl. Valitettavasti niihin ei sisally tavallista yksivarista tayttoa.

Viimeinen valikko sisaltaa piirtely- ja pohjavarin valintataulukot. Valitseminen on toteutettu hyvin mutta tassa tulee suuri puute esille: kerrallaan voi kayttaa vain kolmea piirtovarin pohjavarin lisaksi! Siina suhteessa Koalapainter on voittamaton. Ja kun pastellivaritkaan eivat kuulu valikoimaan: Oh Shit.

Viela kannattaa mainita eras seikka: kun valokynalla piirtelee suoraan ruudulle on siina se huono puoli etta kasi vasahtaa jonkin ajan kuluttua. Lepuuttaminen vastii viela oman aikansa.

Hyvia puolia:

- suuri maara tayttovareja
- paljon vareja valittavana
- luonnollisen tuntuista pirtamista
- valikot katevia

Huo[illegible]suus

Jukka O. Kauppinen alias Grendel/Byterapers on pitkän linjan toimittaja, joka kirjoitti ensimmäisiä tekstejään mekaanisella kirjoituskoneella. JOKin ensimmäinen peliarvostelu julkaistiin 1986, mistä lähtien hän on ahkeroinut tauotta kirjoittaen juttuja muun muassa videopeleistä, tietotekniikasta, viihteestä, ilmailusta ja burleskista. JOKstoriaa-sarjassa pureskellaan niin kotimikroilun kuin peli- ja digijournalismin maailmaa grennulasien läpi nähtynä. jukka@skrolli.fi

Maailman ensimmäinen demokompo

Suomalaisen demopartyhistorian tarina on lähes yhtä vanha kuin skenehistoriankin. Sinne väliin mahtuu monta tarinaa – ja myös monta demokompoa. Kuten se ensimmäinen, vuodelta 1988.

Teksti ja kuvat: Jukka O. Kauppinen alias Grendel / Byterapers, Inc.

Suomessa järjestettiin ennen elokuun 1988 megapartykaaosta kaksi demopartyä: ensimmäinen joulukuussa 1987 (The Jezebels Union & Finnish Gold), toinen keväällä 1988 (Digitize Design Group). Byterapersin Grendelpartyt elokuussa 1988 menivät aivan liian pelottavan monta loikkaa syvään päähän, sillä iisalmelaiseen leirikeskukseen asettui satakunta innokasta demoskenettäjää viikoksi. Kyllä. Viikoksi.

Allekirjoittanut on aina välillä puntaroinut, oliko itsekkäästi nimeämäni Grendelparty oikeasti ensimmäinen, jossa pidettiin demokompot. Ensimmäisillä suomipartyillä niitä ei varmasti ollut. Toisilla partyillä julkaistiin paljon demoja, mutta sumenevat muistikuvat eivät kerro täysin luotettavasti järjestettiinkö siellä kompot vai ei. Aikalaishistoriikit näistä partyista eivät puhu mitään kompoista, joten niihin täytynee luottaa: ei järjestetty.

Grendelpartyjen demoskaban esikuva tuli ulkomailta. Etäisessä Suomessa luettiin tarkoin diskettilehtien partyreportaaseja, ja joskus posti toi swappikavereilta myös paperisten demoskenelehtien kopioita. Minä, **MIY**, **Growl**, **Axa** ja **Gryzor** olimme myös jo käyneet Ruotsissa demopartyillä, kaiketi ensimmäisinä ulkomaan partyillä käyneinä suomiskenereinä. Tuliaisiin lukeutuivat sekä uudet kontaktit, kokemukset että ideoita omille partyille. Mitäs jos siellä pidettäisiin demokompot?

Demoskenen historiankirjoitus on sittemmin osoittanut miten heikolla hapella varhaisen digitaalisen kulttuurin tutkimus voikaan olla, sillä Byterapersin 1988-partyillä järjestettiin kirjallisten lähteiden mukaan maailman vanhimmat demokompot. Mutta kyllä kompoja on aivan taatusti järjestetty jossain aiemminkin, niistä ei vain ole jäänyt minkäänlaista dokumentaatiota.

Mutta näillä mennään. Niinpä Ylä-Savon pikkukaupungin syrjäisellä Koukunjoella tehtiin digitaalista historiaa tuona elokuisena vuoden 1988 päivänä. Demokompoon osallistui nippu Commodore 64 -demoja. Amiga-kisakin olisi järjestetty, mutta osallistujia ei ollut joko tarpeeksi tai lainkaan.

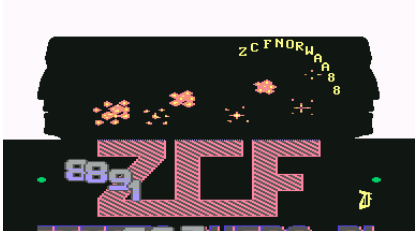
Lets demo party

Pitihän kunnon partyjä muuten pohjustaakin. Norjalainen Academy-demoryhmä teki ystävyyttään pienen kutsudemon suomalaispartyjä varten. Tapahtumaa pohjustettiin myös digitaalisilla letter-tyyppisillä infokirjeillä ja monistetuilla paperikutsuilla, joissa jo mainostettiin tulevaa demokisaa.

Muistikuvat kisasta ovat ainakin kirjoittajalla olemattomat. Demot näytettiin todennäköisesti kuvassa näkyvästä televisiosta ja sitten äänestettiin jotenkin.

Kompon voittanut Finnish Goldin hienoja efektejä ja kaunista musiikkia tarjoillut demo on jo käsitelty aiemmin Skrolli 2018.3:n Skrolleri-palstalla, joten se siitä. Mitä muuta kompossa nähtiin?

2. sija: Z-Circlen *(B)arty demo* oli massiivisuudessaan pitkään omaa luokkaansa. Kokonaisen levypuolen vaatinut 1+9-osainen demo oli megademojen aikakautta edeltänyt mörkäle, jonka osat ladattiin käsin valikon kautta. Automaattista trackloaderia tai osasta toiseen siirtymistä ei kaiketi ollut vielä keksitty. Eihän tämä sinällään ole vielä kummoinen demo, sillä osat ovat irrallisia, ilman yhteistä teemaa, eivätkä järin mieleenpainuviakaan. Niissä oli kuitenkin vahvaa osoitusta Z-Circlen teknisestä osaamisesta, joka vain ei vielä oikein yhdistynyt omaperäiseen designiin tai ideoihin.



3. sija: Zip Cracking Factories tuli partyille Norjasta saakka ja nappasi komean kolmannen sijan neliosaisella demollaan, jota ei muuten näy olevan edes Youtubessa. Täytyypä korjata asia lehden julkaisuun mennessä. Demon osat ovat varsin yksinkertaisia, mutta kolmannen osan efekti- ja tekstivilskeelle täytyy nostaa hattua. Tätä ei epileptikko koodaisi.

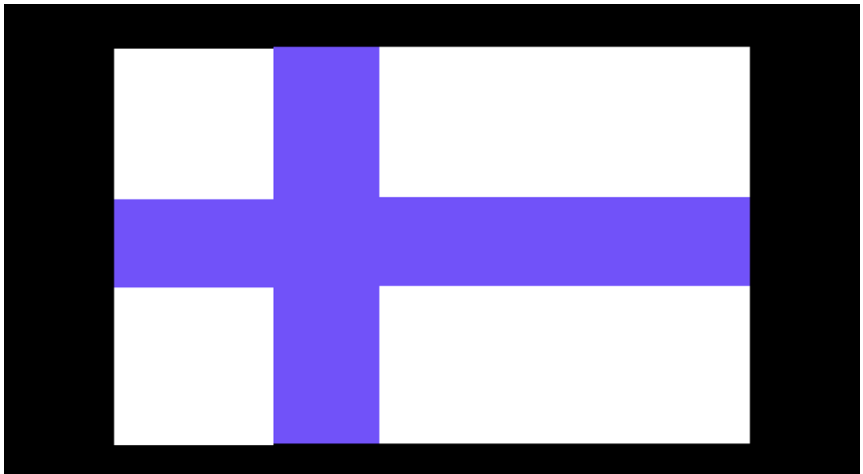
4.The Action Force / Byterapersin demo *In Action!!* näytti bytejen pohjoisen ulottuvuuden osaamista pienessä mutta ei erityisen näyttävässä demossa. TAF nousi kukoistukseen vasta myöhemmin Contex-nimisenä.

5. sijalle yltäneen *Zodux 2 Part I & Part II* -demojen tarina on vähän liian pitkä ja ehkä ilkeäkin tässä käsiteltäväksi. Todettakoon siis, että ne olivat kettuiludemoja Zodux-nimistä, ei erityisen arvostettua ryhmää kohtaan.

7. *Death Star Demossa* ei ole paljoa jälkipolville kommentoitavaa, mutta se tavallaan tiivistää hienosti aikakauden demojen sosiaaliset aspektit. Efektejä ei välttämättä ollut monia tai edes hienoja, mutta voi että skrollitekstejä riitti.

Byterapers Grendelparty 1988:n tulokset (Commodore 64)

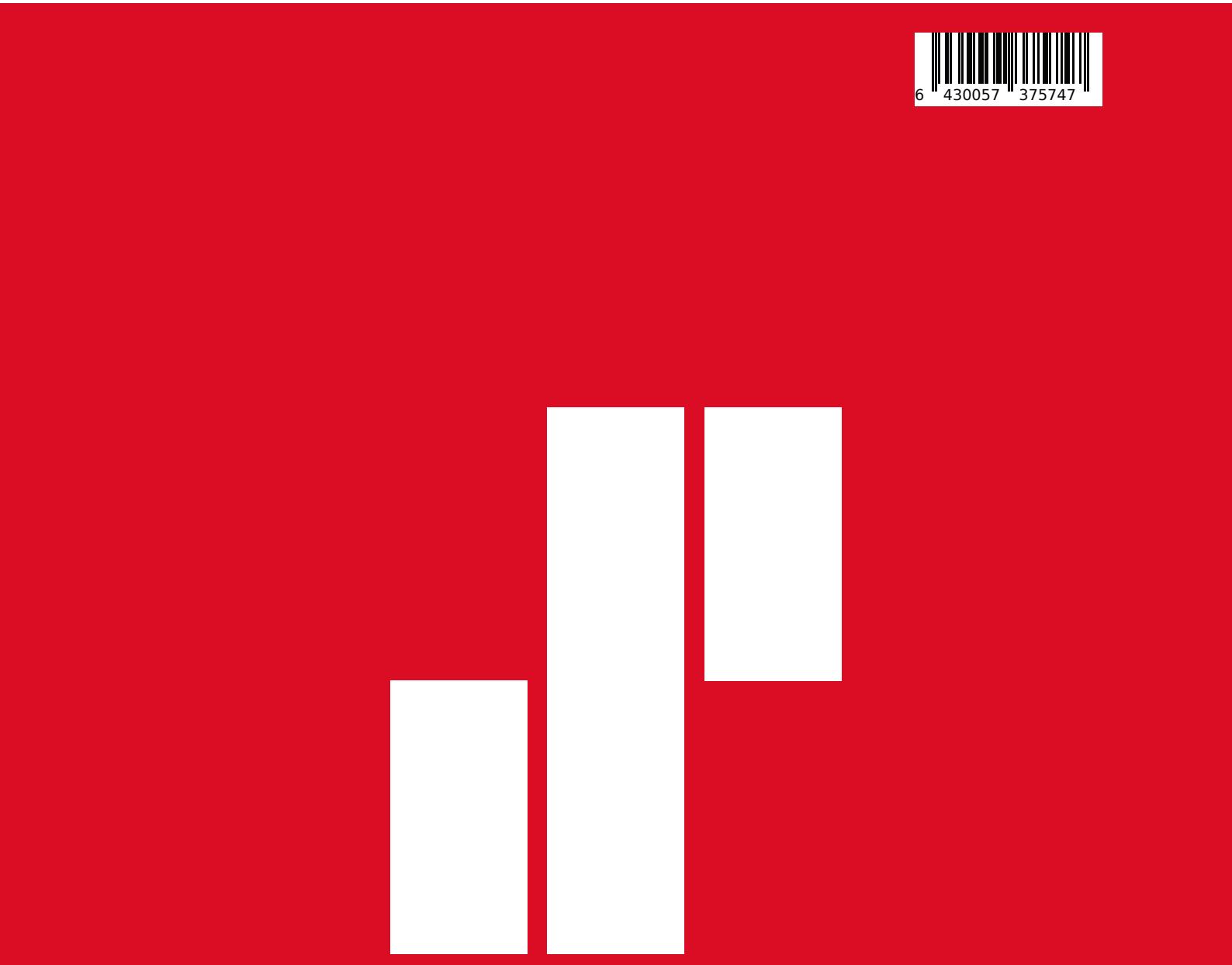
1. Finnish Gold: Contest-demo
2. Z-Circle: (B)arty-demo
3. Zip Cracking Factories: Delirium
4. The Action Force / Byterapers: In Action!!
5. Zodux 2 Part I & Part II
6. Finnish Gold: FIG Again
7. Death Star Demo



JUMISSA

JJ NÄÄS - 19
INDIANA JONES AND THE FATE OF ATLANTIKSEN LÄPIPELUUOHJE ILMESTYI PELIT-LEHDESSÄ ENNEN KUIN AMIGA-VERSIO OLI JULKAISTU.
ÄH! OLISIVAT NYT ODOTTANEET!
PELIT
VIIMEIN:
HUH, 11 DISKETTIÄ. OLISIPA LISÄLEVARI.
PYYSIN SISKOANI PIILOTTAMAAN LEHDEN.
JONNEKIN MISTÄ EN VARMASTI LÖYDÄ.
SELVÄ.
JOITAIN VIIKKOJA MYÖHEMMIN:
EN VAIN TAJUA MITÄ TÄLLÄ "SURVEYOR INSTRUMENTILLA" MUKA PITÄISI TEHDÄ.
TÄÄLLÄHÄN SE LEHTI ON. OLIPAS HIEMAN HELPOSSA PAIKASSA.
KINKKISEMPI TAPAUS OLI CRUISE FOR A CORPSE, JONKA OSTIN VUONNA 1991. OLIN JUMISSA PELIN ALKUPUOLELLA MONTA VUOTTA, VAIKKA PALASIN SEN PARIIN YRITTÄMÄÄN AINAKIN KERRAN KUUSSA.
LUIN HUOLELLA KAIKKIEN PELILEHTIEN VINKKIPALSTAT. TURHAAN.
?
LÄHETINKÖ AVUNPYYNNÖN PELIT-LEHTEEN? EN MUISTA.
"MITÄ CRUISE FOR A CORPSESSA PITÄÄ TEHDÄ KUN PELIN KELLO ON..."
KERRAN SITTEN LATASIN PELIN ALKUPERÄISILTÄ DISKETEILTÄ. SIIHEN ASTI OLIN PELANNUT AINA VARMUUSKOPIODISKETEILTÄ.
DICK SCHMOCKIN DIALOGIVALIKKOON ILMESTYI VAIHTOEHTO, JOTA EI AIEMMIN OLLUT.
AI NÄINKIN VOI KÄYDÄ?
EI TOIMI
TOIMII
Corpse C
SONY
CRUISE FOR A CORPSE
DISK C

AT ZENROBOTICS ROBOTS HAVE OUTNUMBERED THE HUMANS. HELP US RESTORE THE BALANCE.
CHECK OUT ALL OPEN POSITIONS AT HTTPS://CAREERS.ZENROBOTICS.COM
Want to work with groundbraking AI software, smart robots and very talented people – and save the world while doing so? We´re on a mission to solve the world´s resource crisis and we need more brainpower to do it!
ZENROBOTICS®



dev.solita.fi